Swarm and Evolutionary Computation 33 (2017) 18–45

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Review

Noisy evolutionary optimization algorithms – A comprehensive survey

Pratyusha Rakshit[a], Amit Konar[a], Swagatam Das[b]

[a] Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata 700 032, India
[b] Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata 700 108, India

ARTICLE INFO

Keywords:
Evolutionary optimization
Noise
Uncertainty
Sampling
Population sizing
Fitness estimation
Selection

ABSTRACT

Noisy optimization is currently receiving increasing popularity for its widespread applications in engineering optimization problems, where the objective functions are often found to be contaminated with noisy sensory measurements. In absence of knowledge of the noise-statistics, discriminating better trial solutions from the rest becomes difficult in the "selection" step of an evolutionary optimization algorithm with noisy objective/s. This paper provides a thorough survey of the present state-of-the-art research on noisy evolutionary algorithms for both single and multi-objective optimization problems. This is undertaken by incorporating one or more of the five strategies in traditional evolutionary algorithms. The strategies include (i) fitness sampling of individual trial solution, (ii) fitness estimation of noisy samples, (iii) dynamic population sizing over the generations, (iv) adaptation of the evolutionary search strategy, and (v) modification in the selection strategy.

1. Introduction

Real world problems involving system design, control, planning, and scheduling are often formulated in the settings of an optimization problem with an aim to maximize system throughput/efficiency under the constraints on system resources. Typically, a physical process is characterized by a set of measurements and a set of estimators with a mathematical relationship between the measurements and the estimators. For example, in coordinated problem solving, such as box-pushing by twin robots [154], the range data obtained by the robots at any instance of time are the measurements, and the force and/or torque to be developed by the robot for a pre-determined movement of the box are the estimators. The objective functions, here, are energy and time required for local transportation of the box by the robots. The objectives include forces and torques as arguments.

The formulation in the present context is to compositely or independently optimize the two objectives. In single objective formulation, we may simply add the scaled objectives and attempt to optimize the resulting function. In multi-objective formulation, we attempt to optimize the energy and time objectives independently. The problem in the present context is to solve the single/multi-objective optimization problems, when the sensory range measurements are contaminated with noise. Traditional derivative based optimization techniques do not apply to the present problems because of inherent discontinuity of the noisy objectives. The paper addresses evolutionary approach to solve similar *noisy optimization problems* (NOPs).

Evolutionary algorithms (EAs) [16] aim at solving complex optimization problems by mimicking the Darwinian principle of the survival of the fittest [41]. EA commences from an initial population of *trial solutions* uniformly distributed over the search landscape. The trial solutions, representing the potential candidates of the optimization problem, are evolved through an adaptation phase, followed by a competitive selection phase for promotion to the next evolutionary generation. The relative merit of a trial solution is assessed by its corresponding objective function value, often called *fitness*. The 'selection' is an important step in EA as it filters quality solutions (with better fitness measure) from the pool of trial solutions while discarding poor solutions.

Although EA literature has witnessed a radically divergent perspective in solving real-world optimization problems, there is a distinct lack of studies exploring the issues of handling uncertainty in presence of noise. The other forms of uncertainties that might corrupt real-world optimization problems include data incompleteness, inaccuracy in mathematical modelling, environmental condition variation, and infeasible (non-realizable) solutions [42,66,89,98]. Although EA is inherently robust to low levels of noise due to its distributed nature and its non-reliance on gradient information [42], the impact of noise becomes undesirable when it greatly affects the fitness of the trial solutions. The noisy fitness measurements of the trial solutions may adversely distress the performance of selection operation in preserving the true quality solutions over generations in an EA.

Mathematically, the noisy objective function of a trial solution $\overrightarrow{X}$ is represented by

$$f_{noisy}(\overrightarrow{X}) = f(\overrightarrow{X}) + \eta \tag{1}$$

where $f(\overrightarrow{X})$ is the true objective function value and η is the amplitude of the injected noise. It is evident from (1) that due to the noise-induced dynamic variation of the objective surface, the objective function returns different values when repeatedly evaluated for the

http://dx.doi.org/10.1016/j.swevo.2016.09.002
Received 23 May 2016; Received in revised form 26 September 2016; Accepted 29 September 2016
Available online 18 November 2016

same trial solution. In such circumstances, a trial solution of superior quality than the rest of the population may be declined by the selection operation to pass onto the next generation because of its seemingly poor (noisy) fitness estimate. Contrarily, an essentially poor solution with illusively good fitness may deceive the selection process to get accommodation in the next generation [5,15,22,25,26,43,48,118,133–135,137,161,164,192,196].

For the sake of completeness, we here briefly introduce the other forms of uncertainty that might influence the objective function(s) of an optimization problem. They are homologous to noisy optimization, but the approaches to solve them differ from the approaches adopted for noisy optimization. The first homologue, the *robust optimization*, addresses the issue of the perturbation of the design variables (or the parameters of a trial solution) of a real world optimization problem, where the environmental characteristics cannot be retrieved with absolute uncertainty. If a trial solution $\overrightarrow{X}$ is quite susceptible to such variable perturbation $\vec{\delta}$ in its vicinity, the implemented solution may lead to an objective function value $f(\overrightarrow{X} + \vec{\delta})$ significantly different from that of its theoretical measurement $f(\overrightarrow{X})$. Robust optimization aims at minimizing the consequences of such variable perturbations by accentuating the search strategy towards identifying *robust* solutions, capable to confront with the parametric variations of physical systems [24,35,46,60,80,93,109,126,141,142,162,184,185,195,198].

The second optimization problem addresses the uncertainty in truly assessing the objective function value of a trial solution from its approximated *meta-model* [96,97,99,100,114,171]. Mathematically, the approximated objective function of a trial solution $\overrightarrow{X}$ is represented by

$$f_{approx}(\overrightarrow{X}) = f(\overrightarrow{X}) + e(\overrightarrow{X}) \tag{2}$$

where $f(\overrightarrow{X})$ is the true objective function value and $e(\overrightarrow{X})$ represents the approximation error of the meta-model. This is referred to as *fitness approximation*, which attempts to capture the objective function of an optimization problem, where an analytical form of objective function can hardly be formulated or the objective function evaluation is too time costly. The intricacy is then to accurately balance the economical (but erroneous) approximate fitness evaluations and the expensive (but accurate) true fitness evaluations [96,97,99,100,114,171].

The last homologue refers to the *dynamic optimization* problem. It addresses the dynamic variation of the objective surface over time, which may result in shift in the locations of the optima as well. The fitness of a trial solution $\overrightarrow{X}$ is then a function of both $\overrightarrow{X}$ and time t, denoted by $f(\overrightarrow{X}, t)$. The primary challenges for solving such dynamic optimization problems [30,54,81,88,94,117,134,136,144] include (i) efficiently tracking the movement of the optima, (ii) capability to detect newly emerging optima in the search space, and (iii) preserving population diversity to successfully respond to the dynamically changing environment without any restart.

This paper focuses on the first category of uncertainty i.e., the issues of uncertainty management in selecting quality trial solutions in presence of noise in the objective surface(s). Ranking of trial solutions in presence of measurement noise is an open problem in EAs. It requires certain strategies for uncertainty management at each decision-making step to select relatively better solutions from the rest. This paper provides a survey of the various EA-based techniques that have been developed over the past few years for noisy optimization tasks.

Usually, five methodologies are adopted in the literature to attack the problem of noisy evolutionary optimization, including (i) explicit averaging using sampling strategy, (ii) effective fitness estimation of a trial solution from its noisy fitness samples, (iii) implicit averaging using dynamic population sizing, (iv) improved evolutionary search strategies and (v) robust selection. The explicit averaging strategy is concerned with evaluation of noisy objective function of a specific trial solution for a number of times (called *sample size*) and then referring to the aggregated measure (for example, the average) of the samples as its fitness estimate. The strategy is known as *sampling* [2,3,37,39,52,63,68,75,83,92,119,131,143,146,149,150,154–156,173–175,189,202]. The multiple measurements of fitness samples usually have a non-zero variance. The traditional averaging approach, however, presumes uniform probability distribution of all fitness samples and thus fails to capture the local noise distribution effectively. The works in [38,44,104,154–156,167–169,199] proposed alternative approaches to estimate the effective fitness of a trial solution from its noisy fitness samples by proficiently capturing the noise-induced uncertainty in their occurrences.

Evidently, sampling diminishes the impact of noise due to periodically evaluating the objective function for the same trial solution. However, there are evidences [9,11,13,17,21,28,33,63,74,82,139,140,190] that a large sample size with a small population may fail to preserve quality solutions over evolutionary generations due to lack of explorative efficiency. This problem can be circumvented by the third alternative approach to solve noisy optimization. It presumes that the inherent characteristic of an EA to frequently sample the promising area in the search space results in many similar trial solutions in a large population. Hence, the infiltration of noise in the fitness estimate of a trial solution can be implicitly compensated by similar population members.

There exists plethora of research outcomes on improving the evolutionary search strategy to handle noise in optimization problems, especially for multimodal fitness landscape(s). The search strategies are devised to precisely capture the characteristics of the noise-induced fitness landscape by adaptively tuning the control parameters to avoid the misleading direction of noise. Significant researches have been undertaken to adapt the mutation strength of EA to reduce the deceptive effect of noise [8,12,112,163]. The effectiveness of mutation strength adaptation is studied in [23]. The Markov-chain formulation of the migration probability in [121] reveals the potency of high mutation probability to handle the jeopardizing effect of noise. Improved local exploitation strategies are dealt in [45,51] to effectively guide the trial solutions towards the true global optima. On the other hand, the efficacy of the exploration capability to avoid premature convergence in the presence of noise has been studied in [107]. In [77–79], the authors have adopted an *experiential-learning-directed perturbation* strategy to control the movement of the trial solutions based on the information acquired from the last few generations for faster convergence. The research outcome in [124] indicates that a chaotic jump induced controlled movement of trial solutions help them to avoid the deceptive local basins of attraction. Apart from the above-, the works proposed in [20,72,90,105,112,122,152,160,183] need special mentioning.

Finally, the robustness of selection process against noise appears as an important concern of developing a noisy evolutionary optimization algorithm. There exists extensive literature to address the selection step to enhance reliability in filtering true quality solutions from the population even when noise is a predominant factor [36,123,130,188,193]. In [14,32,55,58,65,165,182], statistical hypothesis tests are used to identify the quality trial solution from a pair of competitors based on the measurements of their fitness samples. Probabilistic selection strategies have been adopted in [70,91,191,197] to overcome the deterministic dismissal of true quality solutions due to infiltration of noise. In [71], a *rolling tide* characteristic is induced into a traditional multi-objective optimization to circumvent the deception by noise while selecting optimal solutions. A novel noise-induced uncertainty-handling strategy is proposed in [86,108] with an aim to enhance the robustness of the *rank-based selection* of trial solutions against noise.

The rest of the paper is organized as follows. Section 2 reviews the methods for addressing explicit averaging in detail, followed by a survey on effective fitness estimation strategies in Section 3. The issue of implicit averaging is addressed in Section 4. Section 5 contains a comprehensive description of various methods to improve the evolu-

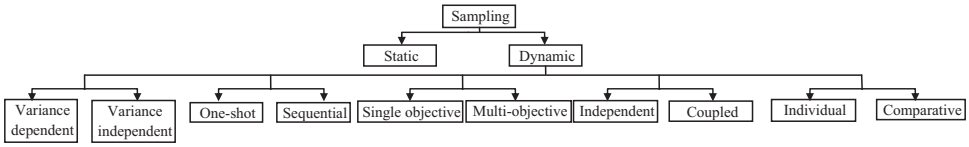

Fig. 1. Classification of sampling strategies for noisy evolutionary optimization.

tionary search strategies in noisy fitness landscape(s). Existing frameworks used to improve selection in noisy evolutionary optimizations are discussed in Section 6. Benchmark problems and performance metrics used to study the comparative analysis of the noisy evolutionary algorithms are listed in Section 7. A discussion on the modality of use of the existing noise handling strategies and open research topics is given in Section 8. Section 9 concludes the paper.

2. Noisy optimization using explicit averaging

The effective fitness estimate of a trial solution $\overrightarrow{X}$ in noisy objective surface is evaluated by

$$\overline{f}(\overrightarrow{X}) = \int_{-\infty}^{\infty} (f(\overrightarrow{X}) + \eta) p(\eta) d\eta \tag{3}$$

where $p(\eta)$ is the probability distribution of noise η contaminating the true objective function value $f(\overrightarrow{X})$. Since an analytical closed form of the effective fitness function in (3) is usually unavailable for most of the noisy optimization problems, it is often approximated using Monte Carlo integration [83].

According to Monte Carlo integration, the noise-induced uncertainty in estimating the fitness $f(\overrightarrow{X})$ of an individual solution $\overrightarrow{X}$ can be reduced by re-evaluating the objective function value $f(\overrightarrow{X})$ several times, often known as *sample size*. The values of multiple *fitness samples* obtained by periodic re-evaluation of $f(\overrightarrow{X})$ are then averaged over to return the mean fitness estimate of $\overrightarrow{X}$. The consequences of sampling strategy on a noisy evolutionary optimization are discussed in [131].

Let the sample size of $f(\overrightarrow{X})$ be n and $f_j(\overrightarrow{X})$ represents the j-th sample of $f(\overrightarrow{X})$. Then using the principle of Monte Carlo integration, the mean fitness estimate $\overline{f}(\overrightarrow{X})$, sample standard deviation $\sigma(f(\overrightarrow{X}))$ and the standard error of the mean fitness $se(\overline{f}(\overrightarrow{X}))$ are calculated as

$$\begin{aligned} \overline{f}(\overrightarrow{X}) &= \tfrac{1}{n}\textstyle\sum_{j=1}^{n} f_j(\overrightarrow{X}) \\ \sigma(f(\overrightarrow{X})) &= \sqrt{\tfrac{1}{n-1}\textstyle\sum_{j=1}^{n}(f_j(\overrightarrow{X}) - \overline{f}(\overrightarrow{X}))^2}. \\ se(\overline{f}(\overrightarrow{X})) &= \tfrac{\sigma(f(\overrightarrow{X}))}{\sqrt{n}} \end{aligned} \tag{4}$$

It is evident from (4) that sampling an individual's objective function n times reduces $se(\overline{f}(\overrightarrow{X}))$ by a factor of $\sqrt{n}$, thus improving the accuracy in the mean fitness estimation. The objective now boils down to optimize the mean fitness estimate. In the noisy minimization problem, introduced in [143], an additional penalty term is also considered in conjunction with the mean fitness estimate. The strategy is realized in simulated annealing (SA) [39,75]. The modified objective function to be minimized is given by

$$f^{\text{mod}}(\overrightarrow{X}) = \overline{f}(\overrightarrow{X}) + 2 \times \alpha(T) \times se(\overline{f}(\overrightarrow{X})) \tag{5}$$

where T is the *simulation temperature*. It is gradually reduced over evolutionary generations. Minimization of the second term guarantees less uncertainty in the fitness assessment. Here $\alpha(T)$ is a weighing parameter which balances the significance of the penalty term with respect to the mean fitness estimate. To sustain the exploration capability of the algorithm in earlier phase while eventually narrowing the search towards global optima, $\alpha(T)$ is set as

$$\alpha(T) = \alpha_0/\beta^T \tag{6}$$

Here α_0 is a small positive value and β is a constant controlling the rate of increase of $\alpha(T)$ with simulation temperature.

Sampling strategies can be primarily classified into two categories based on adaptivity i.e., the precision required by a sampling strategy to know about the quality of a trial solution.

i) **Static sampling (SS):** It allocates equal sample size (i.e., fixed number of re-evaluation of the objective function value) to all population members. However, the sampling requirement of different trial solutions in different regions in the search space may be different. For example, the trial solutions with high fitness variance may require large sample size to improve the accuracy in its fitness estimation, while a quality solution with low fitness variance needs to be assigned with a low sample size to reduce the run-time. Hence, the major shortcoming of the strategy is its incapability to handle the trade-off between the run-time complexity and the computational accuracy.
ii) **Dynamic sampling (DS):** It allocates different sample sizes to individual population members based on their sampling requirements [3,68,146,150,202]. The sampling requirement of a candidate solution can be primarily influenced by one or more of the five vital principles of sample size selection (Fig. 1) including, (a) fitness variance, (b) periodicity in updating sample size, (c) number of objectives, (d) characteristics of the optimization problem and (e) comparative analysis with other population members. Different possible classes of DS strategies are explained next [174].

a) **Variance dependent and independent sampling:** The fitness variance dependent DS strategies allocate sample size to individual solution based on its respective fitness variance. Intuitively, large sample size is allocated to trial solutions with large fitness variance to enhance the quality of their mean fitness estimates. The fitness variance independent DS policies, however, determine the sampling budget of population members irrespective of the knowledge of the fitness variance.
b) **One-shot and sequential sampling:** The one-shot DS strategy estimates the sample budget of an individual solution only for once. However, the one time selection of sample size may not be optimal for simultaneously achieving the optimum evaluation characteristic with reasonable computational complexity. This difficulty can be overcome by sequential DS strategies [37]. Here, after an initial allocation of sample budget, it is verified whether it is capable to achieve the optimal evaluation characteristic. If needed, a fixed number of additional samples of the objective function are periodically evaluated and the optimum criterion is again verified. This is repeated as long as no additional sample is required.
c) **Single and Multi-objective Sampling:** The DS strategy has successfully been utilized in noisy single objective and multi-objective optimization scenarios. Noisy multi-objective optimization algorithms determine the sampling requirements of the trial solutions either based on their individual fitness measures (for example, the Pareto rank in [62]) or an aggregated estimate of all the fitness measures (for example, the reference point distance in [61]). The examples of two cases of multi-objective DS policy are given below.

To jointly optimize multiple objective functions in a multi-objective EA, selection of trial solutions is performed by Pareto ranking [62]. The non-dominated solutions of a population are first

placed into the Pareto front of rank 1. The already ranked members are then temporarily disregard from the current population. The non-dominated members are next identified from the rest of the population and are included into the Pareto front of rank 2. This continues until all population members are assigned with a specific Pareto rank. Evidently, Pareto ranking is induced by the fitness measures of all objective functions for individual trial solutions. The sampling strategy which allocates sample size to a trial solution based on its corresponding Pareto ranking of trial in a multi-objective EA, utilizes all the individual fitness measures of the trial solutions.

Alternatively, the sampling requirement of a trial solution is often found to be influenced by an aggregated estimate of its respective multiple objective function values. For example, in reference point based non-dominated sorting genetic algorithm-II (R-NSGA-II) [61], a reference point $\overrightarrow{R}$ is specified by user in his preferred zones in the multiple fitness landscapes. Let in a multi-objective problem of optimizing N objectives, the objective function values of a trial solution $\overrightarrow{X}$ (evolved through the evolutionary phase) and the user-specified reference point $\overrightarrow{R}$ are respectively given as $\{f_1(\overrightarrow{X}), f_2(\overrightarrow{X}), \ldots, f_N(\overrightarrow{X})\}$ and $\{f_1(\overrightarrow{R}), f_2(\overrightarrow{R}), \ldots, f_N(\overrightarrow{R})\}$. The quality measure of $\overrightarrow{X}$ then can be captured by its distance from $\overrightarrow{R}$ in the multiple fitness landscapes, given by

$$d(\overrightarrow{X}, \overrightarrow{R}) = \sum_{i=1}^{N} w_i |f_i(\overrightarrow{X}) - f_i(\overrightarrow{R})|. \tag{7}$$

Apparently, $d(\overrightarrow{X}, \overrightarrow{R})$ is a single objective aggregated quality measure of multiple objective function values of $\overrightarrow{X}$. Here w_i is a positive weight corresponding to the difference in the i-th fitness measures of $\overrightarrow{X}$ and $\overrightarrow{R}$ for i=[1, N]. Evidently, the sampling strategy determines the sampling requirement of the trial solution $\overrightarrow{X}$ based on the aggregated estimate of it fitness measures, captured by $d(\overrightarrow{X}, \overrightarrow{R})$.

d) **Independent and coupled sampling:** The DS sampling strategy which incorporates the user-specified characteristics of optimization algorithm, is referred to as a coupled sampling policy. For example, the sample size allocation of a trial solution $\overrightarrow{X}$ based on its distance $d(\overrightarrow{X}, \overrightarrow{R})$ from the user-specified reference point $\overrightarrow{R}$ in R-NSGA-II [61] is a coupled sampling stratagem. The independent DS strategy does not depend on the optimization criteria for the selection of the sample size.

e) **Individial and competitive sampling:** Most of the existing DS strategy allocates sample size to an individual $\overrightarrow{X}$ solely based on its own evaluation characteristics, for example, its fitness variance $\sigma^2(f(\overrightarrow{X}))$ or standard error $se(\overline{f}(\overrightarrow{X}))$. However, there are traces of DS policies which allocate sample size based on a comparative study of performance of two individuals in the population. The first category is named as individual sampling while the latter one is often called as competitive sampling. Competitive DS strategy has found its immense application in the selection of quality solutions among two competitors. Let the objective function values of two competitor solutions be contaminated with noise to a large scale (for example, both have large fitness variances). Moreover, let the contenders are closely located in the fitness landscape (for example, their mean fitness estimates are quite similar). In such scenario, it is reruired to resample the objective function values of both the solutions as many number of times until better one is distinguished from the other with high probability. However, if both the competitors are less affected by noise and they are quite separated in the objective space, the better member can be selected with high probability without redundant additional sampling.

Once the sampling requirement of a trial solution is determined based on one (or more) of the aforementioned five criteria, the DS strategy allocates the sample size to the given trial solution based on either a linear or a non-linear functional relationship model between its sampling requirement and its respective sample size. Let $n^{\min}$ and $n^{\max}$ respectively represent the minimum and the maximum sample size. Similarly, the lower and the upper bounds of the sampling requirement are represented by $r^{\min}$ and $r^{\max}$ respectively. The sampling requirement and the sample size of a trial solution $\overrightarrow{X}$ are denoted by $r(\overrightarrow{X})$ and $n(\overrightarrow{X})$ respectively.

According to *linear* DS strategy [174], $n(\overrightarrow{X})$ is set to be proportional with $r(\overrightarrow{X})$ such that

$$n(\overrightarrow{X}) = n^{\min} + (n^{\max} - n^{\min})\frac{r(\overrightarrow{X}) - r^{\min}}{r^{\max} - r^{\min}}. \tag{8}$$

However, the linear setting of relationship between $r(\overrightarrow{X})$ and $n(\overrightarrow{X})$ may require a large value of $n^{\max}$ to effectively capture the uncertainty in the measurement of the fitness samples of $f(\overrightarrow{X})$. This difficulty is circumvented by the *non-linear* sample allocation strategy [174]. There can be three possible design models to control the rate of increment in the sample size with the sampling requirement.

i) **Accelerated non-linear DS:** Here the modulation of sample size $n(\overrightarrow{X})$ with the sampling requirement $r(\overrightarrow{X})$ is given by

$$n(\overrightarrow{X}) = \sqrt[\vartheta]{r(\overrightarrow{X})/r^{\max}}, \quad \vartheta > 1. \tag{9}$$

with ϑ as a positive constant controlling the (accelerated) rate of increase of $n(\overrightarrow{X})$ with $r(\overrightarrow{X})$.

ii) **Decelerated non-linear DS:** The objective of this DS policy is to allocate small sample size, say 10% of $n^{\max}$, to the trial solutions with sampling requirements below 50% while 80% of $n^{\max}$ are reserved for the candidates with sampling requirements as high as 90% of $r^{\max}$. The above necessity can be captured by the following model with ϑ as defined previously

$$n(\overrightarrow{X}) = (r(\overrightarrow{X})/r^{\max})^{\vartheta}, \quad \vartheta > 1 \tag{10}$$

iii) **Combined non-linear DS:** In this case, the mathematical model of relationship between $r(\overrightarrow{X})$ and $n(\overrightarrow{X})$ is designed to utilize the composite benefits of both accelerated and decelerated DS policies. Two such well-known mathematical models include Weibull distribution [101], given as

$$n(\overrightarrow{X}) = 1 - \exp\left(-\frac{1}{\alpha}\frac{r(\overrightarrow{X})}{r^{\max}}\right)^{\beta} \tag{11}$$

and logistic growth [194], given by

$$n(\overrightarrow{X}) = \frac{1}{\left(1 + \exp\left(-\kappa\left(\frac{r(\overrightarrow{X})}{r^{\max}} - \lambda\right)\right)\right)^{\frac{1}{\upsilon}}} \tag{12}$$

Here $\alpha > 1$ and $\beta > 1$ represent the scale parameter and the shape parameter for Weibull distribution, κ denotes the growth rate of $n(\overrightarrow{X})$ with $r(\overrightarrow{X})$ following the logistic growth, λ symbolizes the relative sampling requirement (in %) with the highest growth in the sample size. The occurrence of the maximum growth close to the upper or the lower asymptotes is decided by the parameter $\upsilon > 0$. Both the models ensure a delay (or acceleration) in the sample size allocation at a lower (or higher) sampling requirement. The non-linear DS strategies are pictorially presented in Fig. 2.

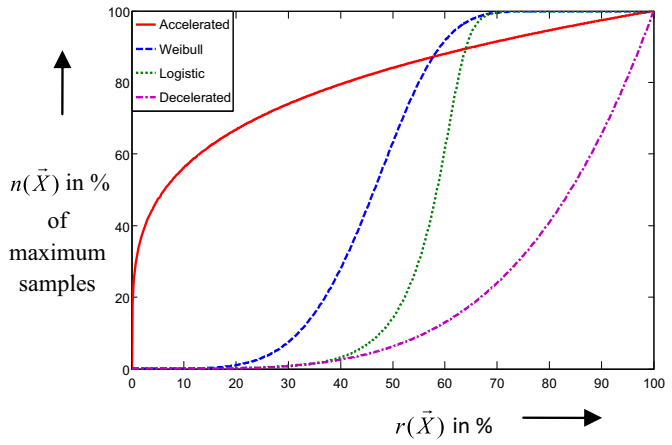

Fig. 2. Non-linear functional relationship between the sampling requirement and the sample size.

We now present some of the well-known sampling strategies.

2.1. Time-based sampling

The *time-based* DS strategy [175] allocates sample size to a trial solution based on the evolutionary generation. In the early exploration phase of evolutionary generations, a larger sample size is preferred for the trial solutions far away from the global optima to capture the uncertainty in their fitness estimation in a relatively liberal way. Thus, it reduces the computational overhead at the early exploration phase. However, the sample size is increased with progress in evolutionary generation to ensure judicious selection of truly quality solutions with good and accurate fitness estimate. The sampling requirement $r(\overrightarrow{X})$ is thus given by

$$r(\overrightarrow{X}) = (G/G_{\max})^{\alpha}, \quad \alpha > 1 \tag{13}$$

where G and $G_{\max}$ represent the current and the maximum number of generations of the evolutionary optimization algorithm. The strategy requires the setting of only one parameter, α, to control the resampling rate. Evidently, here $r^{\min}$=0 and $r^{\max}$=1.

In [176], the time-based sampling is realized in [177] by controlling $r(\overrightarrow{X})$ based on the total number of function evaluations up to current generation, FE_G. However, to enhance the degree of certainty in the objective function values of the final population members, a total number of FE_{final} samples are preserved for the trial solutions in the final population of the evolutionary optimization algorithm [177]. Accordingly, the sampling requirement $r(\overrightarrow{X})$ is thus given by

$$r(\overrightarrow{X}) = \min\left(1, \frac{FE_G}{FE_{total} - FE_{final}}\right)^{\alpha}, \quad \alpha > 1 \tag{14}$$

where FE_{total} represents the total number of function evaluations used as the termination criterion of the evolutionary optimization algorithm. The time-based DS strategy falls under the categories of fitness variance independent, one-shot, independent and individual sampling policies. Moreover, it does not rely on the number of objectives.

Another form of time-based sampling is realized in [176] by allocating the minimum number of samples $n^{\min}$ at the beginning of the optimization algorithm and abruptly increasing the sample size to $n^{\max}$ if the total number of function evaluations up to the current generation FE_G is above a threshold value FE_{Th}. This is captured by modulating $r(\overrightarrow{X})$ as follows.

$$r(\overrightarrow{X}) = \begin{cases} 0, & \text{if} FE_G < FE_{Th} \\ 1, & \text{otherwise} \end{cases} \tag{15}$$

The sampling strategy realized above is referred to as *time-step-based* sampling. However, instead of an abrupt increase in the sample size from $n^{\min}$ to $n^{\max}$, a gradual increment in sample size based on the function evaluations over the evolutionary generations can be realized by setting

$$n(\overrightarrow{X}) = \frac{1}{\left(1 + \exp\left(-\kappa\left(\frac{FE_G - FE_{Th}}{FE_{total} - FE_{final}}\right)\right)\right)^{\frac{1}{\upsilon}}} \tag{16}$$

where $FE_{Th} \in [0,1]$ denotes the point of the highest growth and $\kappa > 0$ denotes the growth rate. The occurrence of the maximum growth close to the upper or the lower asymptotes is decided by the parameter $\upsilon > 0$.

2.2. Domination strength-based sampling

The *domination strength-based* DS strategy [174] allocates sample size to a trial solution $\overrightarrow{X}$ based on the number of solutions it dominates, denoted by $|dom(\overrightarrow{X})|$. Higher the number of solutions $|dom(\overrightarrow{X})|$ dominated by $\overrightarrow{X}$, better is the quality of $\overrightarrow{X}$ and hence more sample size is to be allocated for $\overrightarrow{X}$. A large sample size of such quality solution is valuable for the algorithm to gain accurate knowledge of the noisy fitness landscapes. Contrarily, a small sample size is allocated to a trial solution with a few members being dominated by it. This in turn helps in reducing the computational complexity by avoiding unnecessary re-evaluations of inferior candidates.

Let M denotes the possible maximum number of solutions being dominated by any member of the population P. In other words,

$$M = \max_{\overrightarrow{Z} \in P} |dom(\overrightarrow{Z})|. \tag{17}$$

Apparently $M \geq 0$. Intuitively, the sampling requirement $r(\overrightarrow{X})$ should be set proportional to the size of $dom(\overrightarrow{X})$. In the exploration phase, when $M > 1$, $0 < r(\overrightarrow{X}) < 1$ is set equal to

$$r(\overrightarrow{X}) = \frac{|dom(\overrightarrow{X})|}{M} \quad \text{if } M > 1. \tag{18}$$

It is apparent that the strategy ensures a minimum sampling requirement $r(\overrightarrow{X})$=$r^{\min}$=0 for a trial solution $\overrightarrow{X}$ with $dom(\overrightarrow{X}) = \varphi$. It is worth mentioning that M=0 indicates that all the population members are non-dominated to each other. Hence, all trial solutions are of superior quality and thus their sampling requirement must be maximum, i.e., $r(\overrightarrow{X})$=$r^{\max}$=1.

One more fundamental viewpoint of the strategy is to consider higher sampling budget in the exploitation phase. In the exploitation phase, when the search converges towards the global optima, it is often found that M=1. In this scenario, most of the candidate solutions are expected to be in very close proximity to the Pareto optima. Under such circumstances, even if $|dom(\overrightarrow{X})|$=0, $\overrightarrow{X}$ is assigned with a moderately high $r(\overrightarrow{X})$=0.5 to accurately capture the characteristics of the objective surfaces near the global optima. The aforesaid purpose can be satisfied by setting

$$r(\overrightarrow{X}) = \begin{cases} \frac{|dom(\overrightarrow{X})|}{M}, & M > 1 \\ \max\left(0.5, \frac{|dom(\overrightarrow{X})|}{M}\right), & M = 1 \\ 1, & M = 0 \end{cases} \tag{19}$$

Evidently, the domination strength-based DS strategy falls under the category of comparative sampling, as the sampling requirement of a trial solution is determined based on the number of other solutions it dominates. Additionally, it also satisfies the criteria for fitness variance independent, sequential, truly multi-objective and independent sampling policies.

2.3. Rank-based Sampling

Like domination strength-based sampling, the *rank-based* sampling [174,175] also ensures larger (or smaller) sampling budget for quality (or poor) trial solutions based on their Pareto ranks. In the traditional non-dominated sorting-based multi-objective EA, the non-dominated trial solutions of the current population P are first placed into the Pareto front 1, denoted by $FS(1)$. All the members of $FS(1)$ are assigned with rank 1. The non-dominated members from the rest of the population, i.e., $\{P–FS(1)\}$ are then identified and put into the Pareto front 2, given as $FS(2)$. The ranks of all the candidates of $FS(2)$ are set to 2. The Pareto ranking is continued until each members of the population is assigned with a definite Pareto rank. It is evident that members of Pareto front i (with rank i) dominate the residents of Pareto front j (with rank j). Lower the Pareto rank, better is the quality of a trial solution as it dominates a larger fraction of the current population.

The rank-based DS strategy is thus concerned with allocating a large (or small) sample size for trial solutions in the leading (or last) Pareto fronts with lower (or higher) Pareto ranks. Let $R(\overrightarrow{X})$ be the Pareto rank of a trial solution $\overrightarrow{X}$ and $R_{\max}$ be the maximum Pareto rank achieved by a population member in an evolutionary generation. Then $r(\overrightarrow{X})$ is given by

$$r(\overrightarrow{X}) = \begin{cases} 1 - \left(\frac{R(\overrightarrow{X})-1}{R_{\max}-1}\right)^{\alpha}, & R_{\max} > 1, \quad \alpha > 1 \\ 1, & R_{\max} = 1 \end{cases} \tag{20}$$

Here α is a user-specified parameter, which controls the rate of resampling. The rank-based DS strategy falls under the categories of fitness variance independent, sequential, multi-objective, independent, and comparative sampling policies. An extension of the traditional rank-based sampling is proposed in [175] with an aim to allot additional samples to trial solutions residing in the first M Pareto fronts. The dependence of $r(\overrightarrow{X})$ on $R(\overrightarrow{X})$ is here modeled as

$$r(\overrightarrow{X}) = \begin{cases} 1 - \left(\frac{\min(M,R(\overrightarrow{X}))-1}{\min(M,R_{\max})-1}\right)^{\alpha}, & R_{\max} > 1, \quad \alpha > 1 \\ 1, & R_{\max} = 1 \end{cases} \tag{21}$$

The principal drawback of domination strength-based and rank-based sampling strategies lie in their inability in handling noise in many objective optimization problems where almost all the population members are non-dominated to each other over maximum number of evolutionary generations. This requires a setting of maximum sampling requirement $r^{\max}$=1 (and thus fixed maximum sample size) to most of the population members over every generation, leading to a high computational complexity.

2.4. Standard Error Dynamic Resampling (SEDR)

SEDR [149,150,173], strategy has been employed for solving both noisy single and multi-objective evolutionary optimization problems. The strategy is concerned with the optimal allocation of sampling budget to a trial solution based on the noise strength at its corresponding position in the search space. The contamination level of noise is captured by the standard error of the mean fitness estimate of a trial solution. In single objective scenario, the strategy commences from an initial estimate of the mean fitness value $\overline{f}(\overrightarrow{X})$ of a trial solution $\overrightarrow{X}$, initially assigned with a low sample size $n^{\min}$. Then the strategy periodically re-samples the objective function $f(\overrightarrow{X})$ one more time and re-evaluates the mean $\overline{f}(\overrightarrow{X})$ as well as the standard deviation $\sigma(f(\overrightarrow{X}))$. This is re-iterated until the associated standard error of the mean fitness estimate $se(\overline{f}(\overrightarrow{X}))$ falls below a pre-defined threshold, denoted by se_{th}. Hence, the strategy requires two predefined parameters, $n^{\min}$ and se_{th}. Here the sampling requirement $r(\overrightarrow{X})$ is dependent on the standard error $se(\overline{f}(\overrightarrow{X}))$. The pseudo-code for single objective SEDR strategy for a trial solution $\overrightarrow{X}$ is given next [150].

Procedure single_objective_SEDR

Begin
1. Set $n(\overrightarrow{X})=n^{\min}$.
2. Sample $f(\overrightarrow{X})$ for $n(\overrightarrow{X})$ times.
3. Evaluate the mean fitness estimate $\overline{f}(\overrightarrow{X}) = \frac{1}{n(\overrightarrow{X})}\sum_{j=1}^{n(\overrightarrow{X})} f_j(\overrightarrow{X})$.
4. Calculate the sample standard deviation $\sigma(f(\overrightarrow{X})) = \sqrt{\frac{1}{n(\overrightarrow{X})-1}\sum_{j=1}^{n}(f_j(\overrightarrow{X}) - \overline{f}(\overrightarrow{X}))^2}$.
5. Determine the standard error of the mean fitness estimate $se(\overline{f}(\overrightarrow{X})) = \sigma(f(\overrightarrow{X}))/\sqrt{n(\overrightarrow{X})}$.
6. **If** $se(\overline{f}(\overrightarrow{X})) > se_{th}$
   Then $n(\overrightarrow{X})=n(\overrightarrow{X})+1$ and go to step 2.
   End If.
End.

The strategy is under the categories of fitness variance dependent, sequential, single objective and individual DS strategies. The single objective SEDR strategy has been extended to the multi-objective scenario by averaging the standard error values for all the objectives. The pseudo-code for multi-objective SEDR strategy for a trial solution $\overrightarrow{X}$ with N objectives is presented next [150].

Procedure multi-objective_SEDR

Begin
1. Set $n(\overrightarrow{X})=n^{\min}$.
2. Sample $f_i(\overrightarrow{X})$ for $n(\overrightarrow{X})$ times for i= [1, N].
3. Evaluate the mean fitness estimate $\overline{f}_i(\overrightarrow{X}) = \frac{1}{n(\overrightarrow{X})}\sum_{j=1}^{n(\overrightarrow{X})} f_{i,j}(\overrightarrow{X})$ for i= [1, N].
4. Calculate the sample standard deviation $\sigma(f_i(\overrightarrow{X})) = \sqrt{\frac{1}{n(\overrightarrow{X})-1}\sum_{j=1}^{n}(f_{i,j}(\overrightarrow{X}) - \overline{f}_i(\overrightarrow{X}))^2}$ for i=[1, N].
5. Determine the standard error of the mean fitness estimate $se(\overline{f}_i(\overrightarrow{X})) = \sigma(f_i(\overrightarrow{X}))/\sqrt{n(\overrightarrow{X})}$ for i= [1, N].
6. Evaluate the average standard error $\overline{se}(\overrightarrow{X}) = \frac{1}{N}\sum_{i=1}^{N} se(\overline{f}_i(\overrightarrow{X}))$.
7. **If** $\overline{se}(\overrightarrow{X}) > se_{th}$
   Then $n(\overrightarrow{X})=n(\overrightarrow{X})+1$ and go to step 2.
   End If.
End.

Apparently, the multi-objective SEDR strategy inherits all the characteristics of its single objective counterpart. However, instead of being single objective, it follows the principle of multi-objective DS strategy, influenced by the aggregated measure of the multiple objective function values.

In [178], the threshold value se_{th} is dynamically adjusted based on an additional user-defined parameter $p\in[0,1]$ indicating the level of accuracy in fitness assessment, as given by

$$se_{th}(p) = (1-p)^{\alpha}(se_{th}^{\max} - se_{th}^{\min}) + se_{th}^{\min} \tag{22}$$

where $se_{th}^{\min}$ and $se_{th}^{\max}$ respectively denote the minimum and maximum values of se_{th} and $\alpha > 0$ is a constant parameter controlling the rate of decrease of $se_{th}(p)$ with increase in the need of the accuracy level, preferably towards the convergence of the algorithm (p approaching 1).

2.5. m-Level Dynamic Resampling (mLDR)

Like SEDR, mLDR [149,150,173], strategy has also been used for

handling noise-induced uncertainty in both single and multi-objective evolutionary optimization algorithms. According to the proposed policy, the standard deviation of fitness samples of a trial solution, instead of the standard error of the mean fitness estimate, is used to quantitatively identify the noise strength at the respective location of the given trial solution in the parameter space. The efficiency of the algorithm thus lies in the assignment of optimal sample size to trial solutions based on their respective noise strengths, captured by the corresponding standard deviation of fitness samples. The strategy is very competent to solve NOPs encompassing a finite number (say, m) of discrete intervals of the possible range of the sample standard deviations (representing the noise contamination levels). If the value of m is moderate, it is feasible to design a look-up table comprising m pairs, each including (i) specified intervals of sample standard deviations and (ii) the corresponding sampling budget. A solution is assigned with a sampling budget by first evaluating the solution's sample standard deviation and then searching for the corresponding resampling rate in the look-up table.

Like as in case of SEDR, mLDR also starts with an initial mean fitness estimate of a trial solution, being assigned with a sample size $n^{\min}$. Then the corresponding sample standard deviation $\sigma(f(\overrightarrow{X}))$ is calculated. It is then verified in the look-up table whether the required sample size $n(\sigma(f(\overrightarrow{X})))$ for the calculated standard deviation $\sigma(f(\overrightarrow{X}))$ has been reached by the sample size $n(\overrightarrow{X})$, already assigned to the given trial solution. If not, an additional fitness sample of the candidate is drawn and $\sigma(f(\overrightarrow{X}))$ is updated and the entire procedure of adaptation of sample size is iterated. The process is terminated when the allocated sample size of the objective function is greater than equal to the required sample size. The strategy requires $2m+1$ predefined parameters including, $n^{\min}$, m intervals of sample standard deviation (or noise strength levels), and the corresponding m resampling rates. Here the sampling requirement $r(\overrightarrow{X})$ of a trial solution $\overrightarrow{X}$ is dependent on the sample standard deviation $\sigma(\overline{f}(\overrightarrow{X}))$. The pseudo-code for single objective mLDR strategy for a trial solution $\overrightarrow{X}$ is given next [150].

Procedure single_objective_mLDR

Begin
1. Set $n(\overrightarrow{X})=n^{\min}$.
2. Sample $f(\overrightarrow{X})$ for $n(\overrightarrow{X})$ times.
3. Evaluate the mean fitness estimate $\overline{f}(\overrightarrow{X}) = \frac{1}{n(\overrightarrow{X})}\sum_{j=1}^{n(\overrightarrow{X})} f_j(\overrightarrow{X})$.
4. Calculate the sample standard deviation $\sigma(f(\overrightarrow{X})) = \sqrt{\frac{1}{n(\overrightarrow{X})-1}\sum_{j=1}^{n}(f_j(\overrightarrow{X}) - \overline{f}(\overrightarrow{X}))^2}$.
5. Obtain the required sampling budget $n(\sigma(f(\overrightarrow{X})))$ from the look-up table.
6. **If** $n(\overrightarrow{X}) < n(\sigma(f(\overrightarrow{X})))$
   Then $n(\overrightarrow{X})=n(\overrightarrow{X})+1$ and go to step 2.
   End If.

End.

The sampling requirement of a trial solution $\overrightarrow{X}$ in the multi-objective scenario [150] has been realized by averaging over the sample standard deviations of all N objectives of as given next.

Procedure multi-objective_mLDR

Begin
1. Set $n(\overrightarrow{X})=n^{\min}$.
2. Sample $f_i(\overrightarrow{X})$ for $n(\overrightarrow{X})$ times for i= [1, N].
3. Evaluate the mean fitness estimate $\overline{f}_i(\overrightarrow{X}) = \frac{1}{n(\overrightarrow{X})}\sum_{j=1}^{n(\overrightarrow{X})} f_{i,j}(\overrightarrow{X})$ for i= [1, N].
4. Calculate the sample standard deviation $\sigma(f_i(\overrightarrow{X})) = \sqrt{\frac{1}{n(\overrightarrow{X})-1}\sum_{j=1}^{n}(f_{i,j}(\overrightarrow{X}) - \overline{f}_i(\overrightarrow{X}))^2}$ for i=[1, N].
5. Evaluate the average sample standard deviation $\overline{\sigma}(\overrightarrow{X}) = \frac{1}{N}\sum_{i=1}^{N}\sigma(f_i(\overrightarrow{X}))$.
6. Obtain the required sampling budget $n(\overline{\sigma}(\overrightarrow{X}))$ from the look-up table.
7. **If** $n(\overrightarrow{X}) < n(\overline{\sigma}(\overrightarrow{X}))$
   Then $n(\overrightarrow{X})=n(\overrightarrow{X})+1$ and go to step 2.
   End If.

End.

Evidently, the single and multi-objective mLDR strategies inherit the characteristics of the DR sampling categories, similar to their respective SEDR counterparts.

2.6. Fitness Based Dynamic Resampling (FBDR)

The basic principle of FBDR [150,173] is to assign higher sample size to quality trial solutions to gain accurate knowledge of the promising zones of the fitness landscape contaminated with noise while limiting the sampling budget for inferior candidates as low as possible to reduce the run-time complexity. Like SEDR and mLDR strategies, it is also applied for noisy single and multi-objective scenarios. The pseudo-code is provided next for single objective minimization problem [150].

Procedure single_objective_FBDR

Begin
1. Identify the minimum and maximum possible values of the objective function f, denoted by $f^{\min}$ and $f^{\max}$. Thus evaluate the fitness range $g = f^{\max} - f^{\min}$.
2. Set $n(\overrightarrow{X})=n^{\min}$.
3. Sample $f(\overrightarrow{X})$ for $n(\overrightarrow{X})$ times.
4. Evaluate the mean fitness estimate $\overline{f}(\overrightarrow{X}) = \frac{1}{n(\overrightarrow{X})}\sum_{j=1}^{n(\overrightarrow{X})} f_j(\overrightarrow{X})$.
5. Calculate the normalized fitness estimate $\hat{f}(\overrightarrow{X}) = 1 - \overline{f}(\overrightarrow{X})/g$.
6. Determine the additional sample size $n^{add}(\overrightarrow{X}) = \hat{f}(\overrightarrow{X}) \times (n^{\max} - n^{\min})$
7. Obtain the required sampling budget corresponding to the normalized fitness estimate, $n(\hat{f}(\overrightarrow{X})) = n^{\min} + n^{add}(\overrightarrow{X})$.
8. **If** $n(\overrightarrow{X}) < n(\hat{f}(\overrightarrow{X}))$
   Then $n(\overrightarrow{X})=n(\overrightarrow{X})+1$ and go to step 3.
   End If.

End.

Evidently, for a quality (or inferior) solution $\overrightarrow{X}$ with a low (or high) mean fitness estimate $\overline{f}(\overrightarrow{X})$, the normalized estimate $\hat{f}(\overrightarrow{X}) \in [0, 1)$ is close to 1 (or zero), thus increasing (or decreasing) the sampling requirement. The multi-objective FBDR strategy is proposed in [150], which utilizes the average of the normalized fitness estimates $\tilde{f}(\overrightarrow{X})$ of all objectives to determine the sampling budget of a trial solution $\overrightarrow{X}$. However, for a poor candidate solution $\overrightarrow{X}$ with sufficiently small $\tilde{f}(\overrightarrow{X})$ (for example, close to zero), large sample size allocation only increases the computational complexity without providing necessary information about identifying the global optima in the noisy fitness landscapes. This is circumvented by allocating additional samples only when $\tilde{f}(\overrightarrow{X})$ is above a threshold, say 0.5, by setting

$$n^{add}(\overrightarrow{X}) = (2 \times \max(0.5, \tilde{f}(\overrightarrow{X})))^{\alpha} \times (n^{\max} - n^{\min}) \quad (23)$$

Here α=1 increases $n^{add}(\overrightarrow{X})$ linearly with $\tilde{f}(\overrightarrow{X})$. A setting of $\alpha < 1$ (or $\alpha > 1$) ensures accelerated (or decelerated) growth of $n^{add}(\overrightarrow{X})$ with $\tilde{f}(\overrightarrow{X})$(Fig. 3). Hence the user-defined parameters here include $n^{\min}$, $n^{\max}$ and α.

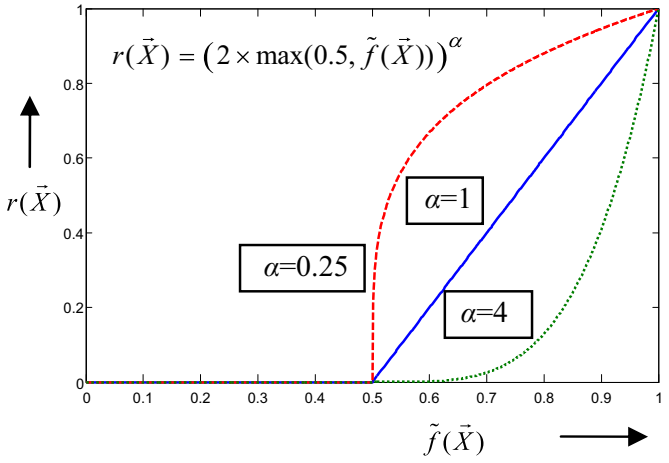

Fig. 3. Effect of α on multi-objective FBDR strategy.

Procedure multi_objective_FBDR

Begin

1. Identify the minimum and maximum possible values of the objective function f_i, denoted by $f_i^{\min}$ and $f_i^{\max}$. Thus evaluate the fitness range $g_i = f_i^{\max} - f_i^{\min}$ for $i = [1, N]$.
2. Set $n(\overrightarrow{X}) = n^{\min}$.
3. Sample $f_i(\overrightarrow{X})$ for $n(\overrightarrow{X})$ times for $i = [1, N]$.
4. Evaluate the mean fitness estimate $\overline{f}_i(\overrightarrow{X}) = \frac{1}{n(\overrightarrow{X})} \sum_{j=1}^{n(\overrightarrow{X})} f_{i,j}(\overrightarrow{X})$ for $i = [1, N]$.
5. Calculate the normalized fitness estimate $\hat{f}_i(\overrightarrow{X}) = 1 - \overline{f}_i(\overrightarrow{X})/g_i$ for $i = [1, N]$.
6. Evaluate the average normalized fitness estimate $\tilde{f}(\overrightarrow{X}) = \frac{1}{N} \sum_{i=1}^{N} \hat{f}_i(\overrightarrow{X})$.
7. Determine the additional sample size $n^{add}(\overrightarrow{X}) = (2 \times \max(0.5, \tilde{f}(\overrightarrow{X})))^{\alpha} \times (n^{\max} - n^{\min})$
8. Obtain the required sampling budget corresponding to the average normalized fitness estimate, $n(\tilde{f}(\overrightarrow{X})) = n^{\min} + n^{add}(\overrightarrow{X})$.
9. **If** $n(\overrightarrow{X}) < n(\tilde{f}(\overrightarrow{X}))$

   Then $n(\overrightarrow{X}) = n(\overrightarrow{X}) + 1$ and go to step 3.

   End If.

End.

It is apparent that the sampling requirement $r(\overrightarrow{X})$ is modulated by $\hat{f}(\overrightarrow{X})$ and $\tilde{f}(\overrightarrow{X})$ for single and multi-objective cases respectively. The strategies are fitness variance independent, sequential, and individual sampling policies.

The minimum and the maximum sample sizes are dynamically modified in [178] based on the user-defined parameter $p \in [0,1]$, denoting the need of accuracy level in the fitness measurement, as follows.

$$n^{\min}(p) = \min(n^{\min}, \lfloor p^{\alpha} \times n^{\min} \rfloor + 1) \tag{24}$$

and

$$n^{\max}(p) = \min(n^{\max}, \lfloor p^{\alpha} \times (n^{\max} - n^{\min} + 1) \rfloor + n^{\min}) \tag{25}$$

where $\alpha > 0$ is a pre-defined constant.

2.7. Hybrid sampling

Several sampling strategies are proposed in literature [150] by hybridizing two different sampling strategies for allocation of optimal sample size. Some of the well-known hybrid schemes are described below.

2.7.1. Rank-time-based sampling

According to the proposed policy in [174], the sample size allocated to a trial solution $\overrightarrow{X}$ is based on the minimum value among the sample size required by rank-based sampling (say, $n^r(\overrightarrow{X})$) and the time-based sampling (say, $n^t(\overrightarrow{X})$) strategies. In other words,

$$n(\overrightarrow{X}) = \min(n^r(\overrightarrow{X}), n^t(\overrightarrow{X})) \tag{26}$$

This strategy thus ensures minimal sample allocation at the earlier exploration phase of the evolutionary generations without sacrificing the quality of solutions at during convergence by accurately identifying the noise-induced uncertainty using larger sample size.

2.7.2. Fitness Based Standard Error Dynamic Resampling (FBSEDR)

FBSEDR strategy synergistically combines the principles of both FBDR and SEDR policies [150,173]. Hence, it assigns the additional sample size to a trial solution $\overrightarrow{X}$ based on its mean fitness estimate $\overline{f}(\overrightarrow{X})$ and the associated standard error $se(\overline{f}(\overrightarrow{X}))$. The terminating condition for single objective SEDR strategy requires $se(\overline{f}(\overrightarrow{X})) > se_{th}$ or $\frac{\sigma(f(\overrightarrow{X}))}{\sqrt{n(\overrightarrow{X})}} > se_{th}$. Evidently, it in turn requires $n(\overrightarrow{X}) > \left(\frac{\sigma(f(\overrightarrow{X}))}{se_{th}}\right)^2$. SEDR and FBDR strategies thus are fused by setting the additional sample size

$$n^{add}(\overrightarrow{X}) = \frac{\hat{f}(\overrightarrow{X}) \times (n^{\max} - n^{\min}) + \left(\frac{\sigma(f(\overrightarrow{X}))}{se_{th}}\right)^2}{2}. \tag{27}$$

The fusion in case of multi-objective scenario is similarly realized by

$$n^{add}(\overrightarrow{X}) = \frac{(2 \times \max(0.5, \tilde{f}(\overrightarrow{X})))^{\alpha} \times (n^{\max} - n^{\min}) + \left(\frac{\sigma(f(\overrightarrow{X}))}{se_{th}}\right)^2}{2}. \tag{28}$$

2.7.3. Fitness Based m-Level Dynamic Resampling (FBmLDR)

FBmLDR strategy utilizes the composite benefits of both FBDR and mLDR policies by setting

$$n^{add}(\overrightarrow{X}) = \frac{\hat{f}(\overrightarrow{X}) \times (n^{\max} - n^{\min}) + n(\sigma(f(\overrightarrow{X})))}{2} \tag{29}$$

in case of noisy single objective scenario [150,173] and

$$n^{add}(\overrightarrow{X}) = \frac{(2 \times \max(0.5, \tilde{f}(\overrightarrow{X})))^{\alpha} \times (n^{\max} - n^{\min}) + n(\overline{\sigma}(f(\overrightarrow{X})))}{2} \tag{30}$$

in case of noisy multi-objective optimization problems [150] respectively.

2.8. Sampling based on fitness variance in local neighborhood

In [154,155,156] the sample size of each trial solution is adapted by the fitness variance in its local neighborhood. The design philosophy relies on an underlying premise that a possible measure of creeping of noise in the neighborhood of a trial solution is anticipated from the fitness variance of a subpopulation around it. A large (or small) fitness variance among the subpopulation members designates a large-scale (or small-scale) detrimental effect of noise in the local neighborhood (of the given trial solution). Under this situation, it is apparent to draw a large (or a small) sample size to accurately estimate the fitness of the given trial solution (or to reduce the run-time without sacrificing the quality of solutions).

Several formulations of the sample size selection can be adopted by sustaining a smaller sample size at lower fitness variance and a larger sample size at larger fitness variance in the local neighborhood of a trial solution. One simple approach could be proportional selection, where the sample size $n(\overrightarrow{X})$ is set proportional to the fitness variance $v(\overrightarrow{X})$

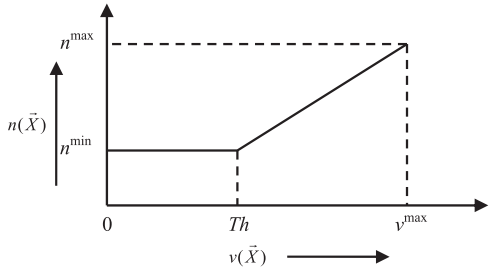

Fig. 4. The linearity used to adapt sample size with fitness variance in local neighborhood.

(above a pre-specified threshold, Th) in the local neighborhood of $\overrightarrow{X}$ [154]. The lower quartile of the fitness variances in the neighborhood of each member in the population is set as the global threshold Th. The selection of lower quartile ensures that sample size should be increased even when the noise-level is one-fourth of its largest possible occurrence. The functional form of relationship (Fig. 4) between $n(\overrightarrow{X})$ and $v(\overrightarrow{X})$ is captured by setting

$$n(\overrightarrow{X}) = \begin{cases} n^{min}, & \text{if} \quad 0 \le v(\overrightarrow{X}) < Th \\ n^{min} + \frac{n^{max} - n^{min}}{n^{max} - Th}(v(\overrightarrow{X}) - Th), & \text{otherwise} \end{cases} \tag{31}$$

where v^{max} represents the maximum fitness variance in the local neighborhoods of all population members in a specific generation.

However, simply setting a proportional law demands a very large $n(\overrightarrow{X})$ at very large $v(\overrightarrow{X})$, which may not be profitable in many contexts, particularly for practical optimization problems. Alternatively, a logistic function may be employed to serve the purpose [155]. As shown in Fig. 5, the relationship between $n(\overrightarrow{X})$ and $v(\overrightarrow{X})$ is a continuous non-decreasing function within $[n^{min}, n^{max}]$

$$v(\overrightarrow{X}) = n^{min} + (n^{max} - n^{min}) \times (1 - \exp(-v(\overrightarrow{X}))) \tag{32}$$

Other forms of non-linear functional relationship between $n(\overrightarrow{X})$ and $v(\overrightarrow{X})$ are proposed in [156,158], respectively given by

$$n(\overrightarrow{X}) = n^{min} + (n^{max} - n^{min}) \times \tanh(v(\overrightarrow{X})) \tag{33}$$

and

$$n(\overrightarrow{X}) = \frac{(n^{max} + n^{min})}{2} + \frac{(n^{max} - n^{min})}{\pi} \times \arctan(v(\overrightarrow{X}) - Th) \tag{34}$$

2.9. Progress-based dynamic sampling

Progress-based DS strategy is proposed in [175] to handle noise in reference point based non-dominated sorting genetic algorithm-II (R-NSGA-II). In traditional NSGA-II, the merged population P_G, comprising both the parent and the child population, are sorted into a number of Pareto fronts according to nondominating criteria. The parent population for the next generation is constructed by selecting the non-dominated sets of solutions according to the ascending order of their Pareto ranking. However, not all trial solutions of a specific lower rank Pareto front may be completely accommodated in the next generation parent population of fixed size. In such partially fitting Pareto front, the trial solutions with higher *crowding distance* measure are prioritized for being promoted to the next generation to maintain population diversity. The diversity metric, realized with crowding distance in traditional NSGA-II, is replaced with reference points of R-NSGA-II. The reference points are defined by the decision maker in the preferred zones in the objective spaces. Trial solutions closer to a reference point possess a higher selection priority. The reference points can be created, adapted, or deleted interactively during the optimization run. The reference points can be of two types, *feasible* and *infeasible*. Feasible reference point is either located on a Pareto front or a trial solution can be found (over generations of R-NSGA-II) to dominate the reference point. Infeasible reference point is neither located in a front nor it can be dominated by any candidate solution.

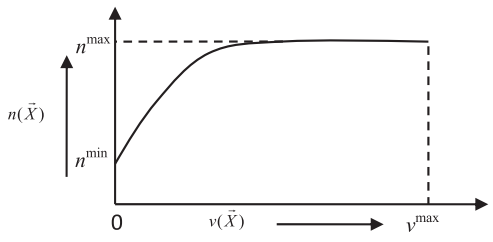

Fig. 5. The nonlinearity used to adapt the sample size with fitness variance in the local neighborhood.

The essence of the progress-based DS strategy is to adapt the sampling requirement of a trial solution based on the progress of the entire population towards a reference point $\overrightarrow{R}$ (irrespective of feasibility criteria) of R-NSGA-II. The average distance of all population members from $\overrightarrow{R}$ is referred to as the progress factor pf. However, due to inherent optimization characteristics of R-NSGA-II, pf is found to be fluctuating over evolutionary generations. A more reliable way is thus to vary the sampling budget with progress factor $\overline{pf}$ averaged over the last g generations. Evidently, a small value of $\overline{pf}$ (i.e., a small distance of the entire population from $\overrightarrow{R}$) indicates the convergence of the population towards the reference point $\overrightarrow{R}$. In such scenario, it is expected to draw more fitness samples for all the candidates of the converged population to extract accurate information of the extent of noise contamination in the preferred zones of the fitness landscapes (i.e., close to $\overrightarrow{R}$). Moreover, to save sampling budget for the population with a large value of $\overline{pf}$, beyond a pre-specified threshold pf^{max}, minimum samples are drawn. The aforementioned principle can be realized by setting

$$r(\overrightarrow{X}) = 1 - \left(\frac{\min(pf^{max}, \overline{pf})}{pf^{max}}\right)^{\alpha} \tag{35}$$

and

$$n(\overrightarrow{X}) = n^{min} + r(\overrightarrow{X}) \times (n^{max} - n^{min}) \tag{36}$$

The performance of the strategy relies on the proper selection of n^{min}, n^{max}, α and pf^{max}. The major inadequacy of the progress-based sampling is that it allocates equal sample size $n(\overrightarrow{X})$ to all population members $\overrightarrow{X}$ of a current generation, as all of them have the same average progress factor $\overline{pf}$. Hence there will be a waste of significant run-time due to assignment of a large sample size to the dominated trial solutions.

2.10. Distance-based dynamic sampling

The inefficiency of the progress-based DS strategy (for balancing the optimal sample allocation among the quality and inferior members of the same population) can be avoided by augmenting the previously mentioned sample allocation scheme with the distance of individual member from the reference point $\overrightarrow{R}$. *Distance-based* DS strategy assigns more samples to trial solutions closer to the preferred reference point. This in turn helps to identify judiciously the true quality solutions in the preferred area in the objective surfaces. According to the proposed strategy, the sampling requirement $r(\overrightarrow{X})$ of a trial solution $\overrightarrow{X}$, with a normalized distance $d(\overrightarrow{X})$ [175] from the reference point $\overrightarrow{R}$, is given as

$$r(\overrightarrow{X}) = (1 - d(\overrightarrow{X}))^{\alpha} \tag{37}$$

where

$$d(\overrightarrow{X}) = \min\left(1, \frac{\delta_{ASF}(\overrightarrow{X})}{D}\right) \tag{38}$$

and $\delta_{ASF}(\overrightarrow{X})$ represents the value of achievement scalarizing function [175] for the given pair of $\overrightarrow{X}$ and $\overrightarrow{R}$. D denotes the maximum reference point distance of the initial population of R-NSGA-II. The strategy also utilizes non-linear transformation functions to determine the sample size $n(\overrightarrow{X})$ corresponding to $r(\overrightarrow{X})$, based on whether the reference point is feasible or infeasible. The proposed strategy also adapts a non-linear transformation function dynamically to estimate $n(\overrightarrow{X})$ from $r(\overrightarrow{X})$ (during run-time of the optimization), based on whether $\overrightarrow{R}$ is feasible or not. However, the feasibility of the reference point is unknown at the beginning of the evolutionary optimization. Hence, at the earlier exploration phases of optimization, the reference point is assumed infeasible and an accelerated non-linear transformation is preferred for sample size allocation. To accomplish the dynamic increment of sample size, two additional criteria are utilized along with the reference point distance.

2.10.1. Progress-based sampling

Intuitively, it is anticipated that the augmentation of the distance-based sample allocation strategy with the progress-based sampling scheme amends the optimal selection of sample size. The hybrid methodology realized with distance-based and progress-based sampling strategies assign a maximum of $n^{\max}$ samples to a certain percentage of best members of the current population based on its progress factor $\overline{pf}$. As $\overline{pf}$ is gradually reduced with the convergence of population, more quality solutions from the population are assigned with maximum sample size. The range of $\overline{pf}$ (per generation in percentage) and the corresponding percentage of population members with the maximally allocated samples is given in Table 1.

The sampling requirement $r(\overrightarrow{X})$ is here modeled as

$$r(\overrightarrow{X}) = \min\left(1, \left(\frac{1 - d(\overrightarrow{X})}{1 - \hat{d}}\right)^{\alpha}\right) \tag{39}$$

where $\hat{d}$ resembles the maximum reference point distance of a trial solution among the selected best population members corresponding to the population with progress factor $\overline{pf}$ [175].

2.10.2. Time-based sampling

In early explorative optimization phase, trial solutions may be trapped at local optima. It is not useful to draw more samples for these locally trapped solutions, even if they are close to the reference point. Rather the noise-induced uncertainty may help them to escape the local optima. The problem can be solved by utilizing distance and progress-based sampling strategy with the time-based allocation scheme. It dynamically changes the transformation functions to increase sample size with an increase in the time elapsed from the beginning of the optimization (Fig. 6).

Evidently, both progress-based and times-based sampling schemes are fitness variance independent, coupled sampling strategies that determine the sample size for each solution individually, perform sequential sampling, use the reference point distance that is an aggregated form of the objective values.

Table 1
Range of $\overline{pf}$ and corresponding percentage of best population members assigned with maximum sample size.

Range of $\overline{pf}$ per generation	Percentage of best population members to be assigned with $n^{\max}$ samples
$10\% \le \overline{pf}$	Only the solution dominating $\overrightarrow{R}$
$5\% \le \overline{pf} < 10\%$	Only the hypothetically best solution closest to $\overrightarrow{R}$
$2.5\% \le \overline{pf} < 5\%$	10%
$1\% \le \overline{pf} < 2.5\%$	20%
$\overline{pf} < 1\%$	40%

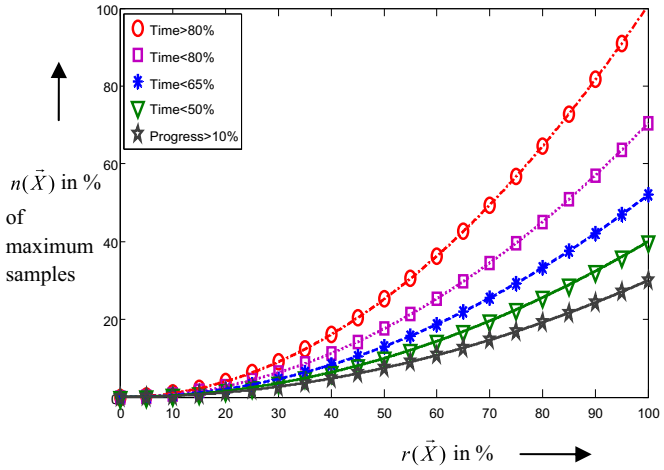

Fig. 6. Non-linear transformation of sampling budget to sample size using distance, progress, and time-based sampling strategies.

2.11. Confidence-based Dynamic Resampling (CDR)

CDR is proposed in [189]. It is a pure multi-objective, sequential, competitive sampling strategy. It also utilizes fitness variance information of two competitive solutions to determine the solution to be resampled further based on Welch confidence interval test [101]. Initially all population members are assigned with an equal sample size, say $n^{\min}$. The mean estimate and sample standard deviation of all objective functions of each candidate solution are then evaluated. Next, the trial solutions are sorted into a number of Pareto fronts according to the non-domination principle based on their mean fitness estimates. However, the so far placement of non-dominated solutions in the same Pareto front may be misleading as the domination relationship between any two population members is derived for same sample size for both of them. Hence, a second order filtering is done in each Pareto front to determine possible true relationship between any two candidates of the sorted population. This is realized here by evaluating the Welch confidence interval value between two contenders $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$, given as

$$\overline{f}_i(\overrightarrow{X_j}) - \overline{f}_i(\overrightarrow{X_k}) \pm t_{df,1-\frac{\alpha}{2N}} \sqrt{\frac{\sigma^2(f_i(\overrightarrow{X_j}))}{n(\overrightarrow{X_j})} + \frac{\sigma^2(f_i(\overrightarrow{X_k}))}{n(\overrightarrow{X_k})}}. \tag{40}$$

for i=[1,N]. Initially $n(\overrightarrow{X_j})$=$n(\overrightarrow{X_k})$=$n^{\min}$. Here α represents the confidence level of a specific Pareto front. To ensure the highest accuracy of identifying the approximate Pareto front in noisy fitness landscapes, α nearly equals to one for competitor solutions, $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$, of Pareto rank 1. The value α is reduced by a step of 0.5 for the succeeding Pareto fronts. N represents the number of objectives of the problem while df denoting the degree of freedom for a paired t-test, given as

$$df = \frac{\left(\frac{\sigma^2(f_i(\overrightarrow{X_j}))}{n(\overrightarrow{X_j})}\right)^2}{n(\overrightarrow{X_j}) - 1} + \frac{\left(\frac{\sigma^2(f_i(\overrightarrow{X_k}))}{n(\overrightarrow{X_k})}\right)^2}{n(\overrightarrow{X_k}) - 1}. \tag{41}$$

If the Welch confidence intervals corresponding to all N objectives of $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$ do not include zero, a statistically significant difference is inferred between the performance of $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$ with the given confidence level α. Hence, the dominating solution or the solution with higher crowding distance is returned.

However, if any of the N intervals for the pair $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$ include zero, it indicates a statistically insignificant discrepancy in their performances and thus requires further resampling to identify the

better one. The solution among $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$ possessing the largest sample standard deviation in the objective f_i ($i \in [1, N]$) with the largest interval including zero (that is, the objective with the largest potential to reduce the undesired insignificance) is the one resampled. After resampling, the Welch confidence intervals are re-evaluated by statistical comparison of all N objectives of $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$. The criterion for the statistical significance of the difference in the performance of $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$ is then again evaluated. This is repeated until the performance difference is proved significant.

However, the increased sample size should be bounded by an upper limit $n^{\max}$. If the resampling of one of the members among $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$, say $\overrightarrow{X_j}$, has already reached $n^{\max}$ but statistically significant performance disparity till not found, it is not further resampled. Instead, the particular objective of the remaining member $\overrightarrow{X_k}$, having the largest sample standard deviation, and the largest interval comprising zero, is resampled. Then the similar iterative processes of calculating confidence intervals and validation of the statistical significance in their performance difference are performed. Once the terminating condition is satisfied, the dominating solution or the solution with higher crowding distance (if the contenders belong to the same Pareto front) is declared as the conqueror.

2.12. Noise analysis selection

Another single objective comparative sampling scenario is found in [47,92,132], where the difference $|\overline{f}(\overrightarrow{X_j}) - \overline{f}(\overrightarrow{X_k})|$ in the fitness estimates of two contenders is compared with $2s$ where s represents the standard deviation of noise jeopardizing the objective surface. The comparison of the fitness difference with $2s$ is justified by considering Gaussian distribution of fitness samples with 95.4% of the samples falling in an interval with a width of $4s$ and centre at the mean value of the distribution [170].

If $|\overline{f}(\overrightarrow{X_j}) - \overline{f}(\overrightarrow{X_k})| > 2s$, the member among $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$ with better fitness is returned as the winner with complete certainty. Otherwise, the statistical comparison indicates that the noise bands of the two solutions significantly overlap. To select the better solution judiciously, farther resampling is required. The normalized measure of the degree of overlap between the two noise bands is given by

$$\Omega = \frac{l + 2s - (h - 2s)}{h + 2s - (l - 2s)} \tag{42}$$

where l=min($\overline{f}(\overrightarrow{X_j}), \overline{f}(\overrightarrow{X_k})$) and h=max($\overline{f}(\overrightarrow{X_j}), \overline{f}(\overrightarrow{X_k})$). Based on Ω, the minimum additional sample size to resample both $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$ for reliable selection is calculated as

$$n^{add} = \left\lceil \left(\frac{1.96}{2 \times (1 - \Omega)} \right)^2 \right\rceil \tag{43}$$

where 1.96 represents the upper critical value of Gaussian distribution with a confidence level of 0.95 [170].

2.13. Optimal Computing Budget Allocation (OCBA)

OCBA technique [50,110,116,145] is concerned with optimal allocation of sample size to each candidate solution to enhance the simulation efficiency with a given computing budget T. In other words, OCBA policy assigns sample sizes to population members by maximizing the probability $p(cs)$ of correct selection of the best trial solution $\overrightarrow{X_{best}}$(such that $\overline{f}(\overrightarrow{X_{best}}) \le \overline{f}(\overrightarrow{X_j})$ for j=[1, NP]) from the population of NP individuals $\overrightarrow{X_1}, \overrightarrow{X_2}, \ldots, \overrightarrow{X_{NP}}$, while satisfying

$$n(\overrightarrow{X_1}) + n(\overrightarrow{X_2}) + \ldots + n(\overrightarrow{X_{NP}}) = T. \tag{44}$$

For minimization problem, the probability of correctly identifying the optimal solution $\overrightarrow{X_{best}}$ is given by

$$p(cs) = p\left(\bigcap_{j=1, X_{best} \neq X_j}^{NP} (\overline{f}(\overrightarrow{X_{best}}) - \overline{f}(\overrightarrow{X_j}) < 0) \right). \tag{45}$$

Considering the Bonferroni inequality [50,145],

$$p\left(\bigcap_{j=1}^{NP} (Y_j < 0) \right) \ge 1 - \sum_{j=1}^{NP} [1 - p(Y_j < 0)], \tag{46}$$

the lower bound of $p(cs)$, referred to as the *approximate probability of correct selection* of $\overrightarrow{X_{best}}$, is given by

$$ap(cs) = 1 - \sum_{j=1, X_{best} \neq X_j}^{NP} p(\overline{f}(\overrightarrow{X_{best}}) > \overline{f}(\overrightarrow{X_j})). \tag{47}$$

The aim of OCBA now can be mathematically formulated as [50,145]

$$\max_{n(\overrightarrow{X_1}), n(\overrightarrow{X_2}), \ldots, n(\overrightarrow{X_{NP}})} \left(1 - \sum_{j=1, X_{best} \neq X_j}^{NP} p(\overline{f}(\overrightarrow{X_{best}}) > \overline{f}(\overrightarrow{X_j})) \right)$$
$$\text{such that } \sum_{j=1}^{NP} n(\overrightarrow{X_j}) = T \text{ and } n(\overrightarrow{X_j}) > 0 \text{ for } j = [1, NP]. \tag{48}$$

Considering normal distribution of fitness samples of trial solutions and for $T \to \infty$, it is shown in [50,145] that $ap(cs)$ can be asymptotically maximized by setting

$$\frac{n(\overrightarrow{X_j})}{n(\overrightarrow{X_k})} = \left(\frac{\sigma(f(\overrightarrow{X_j}))/\delta_{best,j}}{\sigma(f(\overrightarrow{X_k}))/\delta_{best,k}} \right)^2 \tag{49}$$

$$n(\overrightarrow{X_{best}}) = \sigma(f(\overrightarrow{X_{best}})) \sqrt{\sum_{j=1, \overrightarrow{X_{best}} \neq \overrightarrow{X_j}}^{NP} \frac{n^2(\overrightarrow{X_j})}{\sigma^2(f(\overrightarrow{X_j}))}} \tag{50}$$

for j, $k \in \{1, 2, \ldots, NP\}$, $\overrightarrow{X_j} \neq \overrightarrow{X_k} \neq \overrightarrow{X_{best}}$ and $\delta_{best,j} = \overline{f}(\overrightarrow{X_j}) - \overline{f}(\overrightarrow{X_{best}})$.

The OCBA technique, described next, is a sequential, truly multi-objective, fitness variance dependent, comparative sampling policy.

Procedure OCBA

Begin

1. Set the index for evolutionary index G=0 and sample $f(\overrightarrow{X_j}(G))$ for $n(\overrightarrow{X_j}(0))$=$n^{\min}$ times for j= [1, NP].
2. Perform additional Δ simulations for $f(\overrightarrow{X_j}(G))$ and thus evaluate new values of $\sigma(\overrightarrow{X_j}(G))$ and $\delta_{best,\,j}$ for j=[1, NP].
3. Determine $n(\overrightarrow{X_j}(G+1))$ using the evaluated values of $\sigma(\overrightarrow{X_j}(G))$ and $\delta_{best,\,j}$ to satisfy OCBA conditions (1) and (2) for j=[1, NP].
4. Perform $\max(0, n(\overrightarrow{X_j}(G+1)) - n(\overrightarrow{X_j}(G)))$ additional simulations for $f(\overrightarrow{X_j}(G))$ for j=[1, NP].
5. Set $G \leftarrow G+1$.
6. **If** $\sum_{j=1}^{NP} n(\overrightarrow{X_j}(G)) < T$
   Then go to step 2.
   End If.

End.

In [49,179], OCBA technique is used to find the optimal sampling budget of the population members to find the best m trial solutions out of the NP population members. It is realized by partitioning the entire population P into two sub-populations, including S_m consisting of the m best solutions and the rest as P/S_m. For minimization problem, the probability of correctly identifying m optimal solutions is given by

$$p(cs_m) = p\left(\cap_{\overrightarrow{X_j} \in S_m} \cap_{\overrightarrow{X_k} \notin S_m} (\overline{f}(\overrightarrow{X_j}) \le \overline{f}(\overrightarrow{X_k})) \right)$$
$$= p\left(\max_{\overrightarrow{X_j} \in S_m} \overline{f}(\overrightarrow{X_j}) \le \min_{\overrightarrow{X_k} \notin S_m} \overline{f}(\overrightarrow{X_k}) \right) \tag{51}$$

For a constant threshold c, $p(cs_m)$ can be approximated as

$$\begin{aligned} p(cs_m) &= p\left(\cap_{\overrightarrow{X_j}\in S_m}\cap_{\overrightarrow{X_k}\notin S_m}(\overline{f}(\overrightarrow{X_j}) \le \overline{f}(\overrightarrow{X_k}))\right) \\ &\ge p(\overline{f}(\overrightarrow{X_j}) \le c \text{ and } \overline{f}(\overrightarrow{X_k}) \ge c,\ \overrightarrow{X_j} \in S_m \text{and} \overrightarrow{X_k} \notin S_m). \\ &= \prod_{\overrightarrow{X_j}\in S_m} p(\overline{f}(\overrightarrow{X_j}) \le c) \prod_{\overrightarrow{X_k}\notin S_m} p(\overline{f}(\overrightarrow{X_k}) \ge c) \end{aligned} \tag{52}$$

The OCBA-m algorithm can now be formulated as

$$\begin{aligned} &\max_{n(\overrightarrow{X_1}),n(\overrightarrow{X_2}),\ldots,n(\overrightarrow{X_{NP}})} \prod_{\overrightarrow{X_j}\in S_m} p(\overline{f}(\overrightarrow{X_j}) \le c) \prod_{\overrightarrow{X_k}\notin S_m} p(\overline{f}(\overrightarrow{X_k}) \ge c) \\ &\text{such that } \sum_{j=1}^{NP} n(\overrightarrow{X_j}) = T \text{ and } n(\overrightarrow{X_j}) > 0 \text{ for } j = [1, NP]. \end{aligned} \tag{53}$$

The Lagrangian relaxation of (53) is given by

$$L = \prod_{\overrightarrow{X_j}\in S_m} p(\overline{f}(\overrightarrow{X_j}) \le c) \prod_{\overrightarrow{X_k}\notin S_m} p(\overline{f}(\overrightarrow{X_k}) \ge c) - \lambda\left(\sum_{j=1}^{NP} n(\overrightarrow{X_j}) - T\right) \tag{54}$$

where λ is the Lagrange multiplier. The Karush–Kuhn–Tucker (KKT) conditions of (54) are given by

$$\begin{aligned} &\frac{\partial L}{\partial n(\overrightarrow{X_j})} = 0, \quad \text{for } \overrightarrow{X_j} \in S_m \\ &\frac{\partial L}{\partial n(\overrightarrow{X_k})} = 0, \quad \text{for } \overrightarrow{X_k} \notin S_m \\ &\frac{\partial L}{\partial \lambda} = 0 \end{aligned} \tag{55}$$

Considering normal distribution of fitness samples and $\delta_j = \overline{f}(\overrightarrow{X_j}) - c$ for j=[1, NP], we can have

$$\begin{aligned} &p(\overline{f}(\overrightarrow{X_j}) \le c) \\ &= \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi}\left(\sigma(f(\overrightarrow{X_j}))/\sqrt{n(\overrightarrow{X_j})}\right)} \exp\left(-\frac{(x-\delta_j)^2}{2(\sigma^2(f(\overrightarrow{X_j}))/n(\overrightarrow{X_j}))}\right)dx \\ &= \int_{\frac{\delta_j}{\left(\sigma(f(\overrightarrow{X_j}))/\sqrt{n(\overrightarrow{X_j})}\right)}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right)dt \end{aligned} \tag{56}$$

and

$$\begin{aligned} &p(\overline{f}(\overrightarrow{X_k}) \ge c) \\ &= \int_{0}^{\infty} \frac{1}{\sqrt{2\pi}\left(\sigma(f(\overrightarrow{X_k}))/\sqrt{n(\overrightarrow{X_k})}\right)} \exp\left(-\frac{(x-\delta_k)^2}{2(\sigma^2(f(\overrightarrow{X_k}))/n(\overrightarrow{X_k}))}\right)dx \\ &= \int_{-\frac{\delta_k}{\left(\sigma(f(\overrightarrow{X_k}))/\sqrt{n(\overrightarrow{X_k})}\right)}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right)dt \end{aligned} \tag{57}$$

Moreover, let,

$$n(\overrightarrow{X_j}) = \alpha_j T \text{ and } \sum_{j=1}^{NP} \alpha_j = 1. \tag{58}$$

Considering $T\to\infty$ and combining (54)–(58), we have

$$\frac{n(\overrightarrow{X_j})}{n(\overrightarrow{X_k})} = \frac{\alpha_j}{\alpha_k} = \left(\frac{\sigma(f(\overrightarrow{X_j}))/\delta_j}{\sigma(f(\overrightarrow{X_k}))/\delta_k}\right)^2 \tag{59}$$

To find the optimal setting of the threshold c, the trial solution in S_m and P/S_m are arranged in the ascending order of their effective fitness estimates (for minimization problem). Let the trial solutions at the bottom and top positions of S_m and P/S_m are respectively denoted by $\overrightarrow{X_m}$ and $\overrightarrow{X_{m+1}}$. Intuitively, the optimal value of c is identified by maximizing both $c - \overline{f}(\overrightarrow{X_m})$ and $\overline{f}(\overrightarrow{X_{m+1}}) - c$, which yields [179],

$$c = \frac{\frac{\sigma(f(\overrightarrow{X_{m+1}}))}{\sqrt{n(\overrightarrow{X_{m+1}})}}\overline{f}(\overrightarrow{X_m}) + \frac{\sigma(f(\overrightarrow{X_m}))}{\sqrt{n(\overrightarrow{X_m})}}\overline{f}(\overrightarrow{X_{m+1}})}{\sigma(f(\overrightarrow{X_m})) + \sigma(f(\overrightarrow{X_{m+1}}))} \tag{60}$$

Table 2
Comparison of non-adaptive sampling rules in D-dimensional search space.

Sampling rule	Sampling requirement at generation G
Linear	G
Quadratic	G^2
Cubic	G^3
2exp	2^G
exp/10	$\lceil \exp(G/10) \rceil$
math1	$\lceil \exp(4G/5D)/D^2 \rceil$
math2	$\lceil \exp(G/10)/D^2 \rceil$
math3	$\lceil \exp(G/\sqrt{D})/D^2 \rceil$

The complexity of different sampling methodologies has been studied in [5,48,118,133–135]. Eight different non-adaptive sampling rules are compared in [118] based on the number of samples of trial solutions in generation G as given in Table 2. The performance of 2*exp* and *polynomial* functions gradually deteriorate with increase and decrease in search space dimensions respectively. Contrarily, *math*2 remains successful in attaining the near-optimal solution in both situations. In [5], two additive noise models are considered, including the Gaussian noise and heavy tailed noise (with finite variance), to examine the contamination effect on the run-time of the algorithm. As evident from Table 3, the run-time of the algorithm with Gaussian noise-corrupted fitness measures differs from the one in the absence of noise by an additional logarithmic factor. However, in the presence of the heavy tail additive noise, the run-time is quadratic on the run-time in the noise-free case.

3. Fitness estimation

Because of sampling, we have multiple fitness measurements of a given trial solution. These measurements in general take different values with a nonzero variance. Traditional noisy evolutionary algorithms estimate the true fitness of a given trial solution by averaging over its noisy fitness samples [186]. However, this averaging strategy presumes equal probability of occurrence of all fitness samples, thereby offering a poor fitness estimate in most of the real situations. In [127], it is shown that noise does not always follow normal or Gaussian or other central distribution, for which averaging could be a good estimate. Under such situations, the kurtosis and skewness are shown to be two important parameters to describe the stochastic nature of noisy fitness samples. Alternative approaches of estimating fitness of a trial solution, considering non-uniform distribution of its fitness samples, are proposed in [38].

Table 3
Comparison of run-times of different problems solved by (1+1) EA.

Problem	Run-time (Noise-Free)	Run-time in presence of	
		Gaussian Noise	Heavy Tail Noise
OneMax	$O(D \log D)$	$O(D (\log D)^2)$	$O(D^2 (\log D)^2)$
Max Clique	$O(D^5)$	$O(D^5 \log D)$	$O(D^{10})$
Sorting	$O(D^2 \log D)$	$O(D^2 (\log D)^2)$	$O(D^4 (\log D)^2)$

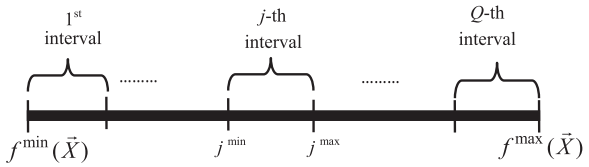

Fig. 7. Fitness intervals in the sample space.

3.1. Expected fitness estimation using uniform fitness interval

The proposed strategy [154], is concerned with the evaluation of the expected value of the samples of $f(\overrightarrow{X})$, symbolized by $E(f(\overrightarrow{X}))$. To accomplish the strategy, the minimum and the maximum value of the observed samples are first respectively recorded as $f^{\min}(\overrightarrow{X})$ and $f^{\max}(\overrightarrow{X})$. Now the entire range $[f^{\min}(\overrightarrow{X}), f^{\max}(\overrightarrow{X})]$ is divided into Q intervals of equal length as shown in Fig. 7. The number of samples residing in the j-th interval (specified by $[j^{\min}, j^{\max}]$) is denoted by c_j for j=[1, Q]. Then the probability of occurrence of fitness samples in the j-th interval, p_j, is calculated as follows.

$$p_j = c_j/n(\overrightarrow{X}) \text{ for } j = [1,\ Q]. \tag{61}$$

Representing $\overline{f}^j(\overrightarrow{X})$, the average fitness of $\overrightarrow{X}$ in the j-th interval, for j=[1, Q] by the centre of the same interval, i.e.,

$$\overline{f}^j(\overrightarrow{X}) = (j^{\min} + j^{\max})/2, \tag{62}$$

we obtain $E(f(\overrightarrow{X}))$ as

$$E(f(\overrightarrow{X})) = \sum_{j=1}^{Q} p_j \times \overline{f}^j(\overrightarrow{X}). \tag{63}$$

3.2. Expected fitness estimation using non-uniform fitness interval

The uniform partition of sample space, being independent of sample distribution, may not be capable to identify the true probability of occurrence of fitness samples suitably over a wide space. An alternative approach is given in [155]. Instead of dividing the sample space $[f^{\min}(\overrightarrow{X}), f^{\max}(\overrightarrow{X})]$ in pre-specified Q intervals, the proposed strategy deals with non-uniformly partitioning the sample space based on the fitness variance $\sigma^2(f(\overrightarrow{X}))$.

First, the entire range $[f^{\min}(\overrightarrow{X}), f^{\max}(\overrightarrow{X})]$ is divided into two intervals of equal length. The resulting intervals are represented by $[f^{\min}(\overrightarrow{X}), f^{\text{mid}}(\overrightarrow{X})]$ and $[f^{\text{mid}}(\overrightarrow{X}), f^{\max}(\overrightarrow{X})]$, respectively where $f^{\text{mid}}(\overrightarrow{X}) = (f^{\min}(\overrightarrow{X}) + f^{\max}(\overrightarrow{X}))/2$. If the variance of the fitness samples lying in the first interval is found to be greater than $\sigma^2(f(\overrightarrow{X}))/n(\overrightarrow{X})$, it is again subdivided into two more equal intervals, represented by $[f^{\min}(\overrightarrow{X}), f^{\text{mid},1}(\overrightarrow{X})]$ and $[f^{\text{mid},1}(\overrightarrow{X}), f^{\max}(\overrightarrow{X})]$, respectively, where $f^{\text{mid},1}(\overrightarrow{X}) = (f^{\min}(\overrightarrow{X}) + f^{\text{mid}}(\overrightarrow{X}))/2$. The same approach is used in the second interval also. The same procedure is repeated for all subsequent intervals until the variance of the fitness samples in each interval falls below $\sigma^2(f(\overrightarrow{X}))/n(\overrightarrow{X})$. Consequently, the entire sample space $[f^{\min}(\overrightarrow{X}), f^{\max}(\overrightarrow{X})]$ is now divided into Q (not pre-specified) intervals of unequal length as indicated in Fig. 8.

It is evident from the proposed nonuniform partitioning of the sample space that the fitness samples in the longer intervals are rare

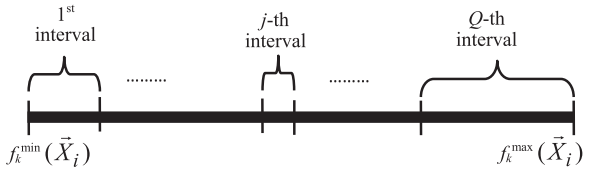

Fig. 8. Fitness intervals in the sample space.

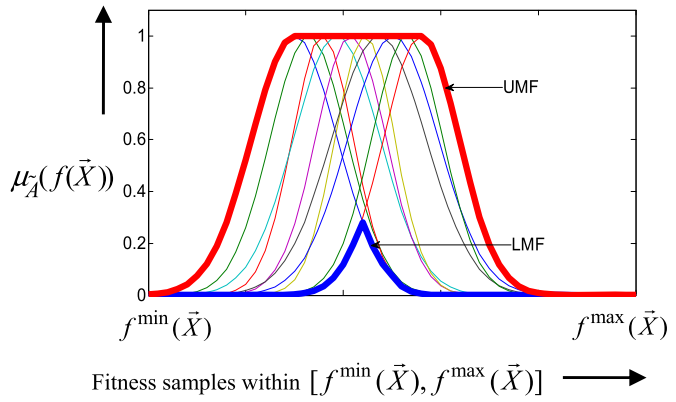

Fig. 9. IT2FS formation from primary memberships of fitness samples lying within the range $[f^{\min}(\overrightarrow{X}), f^{\max}(\overrightarrow{X})]$.

samples, probably owing to noise. Hence, their contribution to the true fitness estimate of $\overrightarrow{X}$ should not be regarded similarly to that of the fitness samples in the smaller interval. We measure the contribution of the fitness samples of the interval j to the fitness estimate of the trial solution $\overrightarrow{X}$ by the probability measure

$$p_j = c_j/n(\overrightarrow{X}) \tag{64}$$

where c_j represents the number of samples of $f(\overrightarrow{X})$ in the jth interval for j=[1, Q]. Let $\hat{f}^j(\overrightarrow{X})$ be the median value of the samples of $f(\overrightarrow{X})$ in the j-th interval, for j=[1, Q]. The expected fitness estimate is then obtained by

$$E(f(\overrightarrow{X})) = \sum_{j=1}^{Q} p_j \times \hat{f}^j(\overrightarrow{X}). \tag{65}$$

3.3. Centroid fitness estimation using interval type-2 fuzzy set

An *interval type-2 fuzzy set* (IT2FS) model is employed in [156] [159] to model the noise-induced uncertainty in the occurrence of fitness samples. First, the sample space is divided into Q intervals of equal length (as done in case of uniform partitioning). The $f(\overrightarrow{X})$ in each interval is represented by a Gaussian type *membership function* (MF) with mean and variance equal to the respective mean and variance of the fitness samples in the selected interval. An IT2FS $\mu_{\tilde{A}}(f(\overrightarrow{X}))$ is constructed (Fig. 9) taking union of all such Q type-1 Gaussian MFs, each one for one interval. The *footprint of uncertainty* (FOU) of $\mu_{\tilde{A}}(f(\overrightarrow{X}))$ represents the possible uncertainty in the sample space of $f(\overrightarrow{X})$. The FOU is bounded by two curves, *upper membership function* (UMF) $\overline{\mu}_{\tilde{A}}(f(\overrightarrow{X}))$ and *lower membership function* (LMF) $\underline{\mu}_{\tilde{A}}(f(\overrightarrow{X}))$[125]. The UMF is then approximated as a flat-top MF to maintain convexity characteristics.

IT2 defuzzification is then performed to determine the left and right end-point centroids [125], C_l and C_r, respectively, using

$$C_l = \frac{\sum_{m=1}^{l} z_m \cdot \overline{\mu}_{\tilde{A}}(z_m) + \sum_{m=l+1}^{q} z_m \cdot \underline{\mu}_{\tilde{A}}(z_m)}{\sum_{m=1}^{l} \overline{\mu}_{\tilde{A}}(z_m) + \sum_{m=l+1}^{q} \underline{\mu}_{\tilde{A}}(z_m)}$$
$$C_r = \frac{\sum_{m=1}^{r} z_m \cdot \underline{\mu}_{\tilde{A}}(z_m) + \sum_{m=r+1}^{q} z_m \cdot \overline{\mu}_{\tilde{A}}(z_m)}{\sum_{m=1}^{r} \underline{\mu}_{\tilde{A}}(z_m) + \sum_{m=r+1}^{q} \overline{\mu}_{\tilde{A}}(z_m)} \tag{66}$$

where l (or r) represents the left (or the right) switching point from UMF to LMF (or from LMF to UMF) for q distinct points in the interval $[f_k^{\min}(\overrightarrow{X_i}), f_k^{\max}(\overrightarrow{X_i})]$. Here z_m symbolizes the m-th sampled point in the same interval with z_1=$f^{\min}(\overrightarrow{X})$ and z_q=$f^{\max}(\overrightarrow{X})$. The centroid fitness $\tilde{f}_c(\overrightarrow{X})$, obtained by

$$\tilde{f}_c(\overrightarrow{X}) = (C_l + C_r)/2, \tag{67}$$

is then referred to as the true estimate of $f(\overrightarrow{X})$.

3.4. Memory based fitness evaluation

The approaches proposed in [104,167,168,169], are concerned with evaluating the effective fitness estimate of a trial solution not only based on its own *sampled fitness*, but also on the sampled fitness of the remaining population members. This information is stored in a memory, which is referred to as *explicit memory* or *history of search*. Let $\overrightarrow{X}$ be a trial solution containing H population members $\{\overrightarrow{X_j}\}_{j=1}^{H}$ in its history of search. It is worth mentioning that the trial solution $\overrightarrow{X}$ is also included in its history of search with $\overrightarrow{X_1} = \overrightarrow{X}$. The true and sampled fitness (in noisy environment) of each $\overrightarrow{X_j}$ are respectively represented by $f(\overrightarrow{X_j})$ and $f_{noisy}(\overrightarrow{X_j})$ for j=[1, H]. Let d_j represents the distance of $\overrightarrow{X_j}$ from $\overrightarrow{X}$ in the parameter space for j=[1, H]. According to the proposed approach in [167], $f(\overrightarrow{X_j})$ is normally distributed around $f(\overrightarrow{X})$ with a variance kd_j, denoted by

$$f(\overrightarrow{X_j}) \sim N(f(\overrightarrow{X}), kd_j) \tag{68}$$

Similarly, $f_{noisy}(\overrightarrow{X_j})$ is assumed to be distributed normally with mean $f(\overrightarrow{X})$ and variance $kd_j+\sigma^2$ for zero mean normal distribution of additive noise η of variance σ^2. Symbolically,

$$f_{noisy}(\overrightarrow{X_j}) = f(\overrightarrow{X_j}) + \eta \sim N(f(\overrightarrow{X}), kd_j + \sigma^2). \tag{69}$$

Hence, the probability density function of $f_{noisy}(\overrightarrow{X_j})$ is given by

$$p\left(f_{noisy}(\overrightarrow{X_j}), d_j\right) = \frac{1}{\sqrt{2\pi(kd_j + \sigma^2)}} \exp\left(-\frac{\left(f_{noisy}(\overrightarrow{X_j}) - f(\overrightarrow{X_j})\right)}{2(kd_j + \sigma^2)}\right). \tag{70}$$

The probability of obtaining the fitness estimate of $\overrightarrow{X}$ jointly based on the sampled fitness values of the H members in its history of search is represented by

$$L = \prod_{j=1}^{H} p\left(f_{noisy}(\overrightarrow{X_j}), d_j\right). \tag{71}$$

Treating L as the likelihood with respect to $f(\overrightarrow{X})$, the effective fitness estimate, $\widetilde{f}_H(\overrightarrow{X})$, can be obtained by maximizing L for $f(\overrightarrow{X})$. It then returns $\widetilde{f}_H(\overrightarrow{X})$ as a weighted average as its own samples fitness and the sampled fitness measurements of the other members in its history of search, given by

$$\widetilde{f}_H(\overrightarrow{X}) = \frac{\sum_{j=1}^{H} \frac{f_{noisy}(\overrightarrow{X_j})}{kd_j + \sigma^2}}{\sum_{j=1}^{H} \frac{1}{kd_j + \sigma^2}} = \frac{f_{noisy}(\overrightarrow{X}) + \sum_{j=1}^{H} \frac{\sigma^2}{kd_j + \sigma^2} f_{noisy}(\overrightarrow{X_j})}{1 + \sum_{j=1}^{H} \frac{\sigma^2}{kd_j + \sigma^2}}. \tag{72}$$

The proposed approach, however, faces difficulty in truly identifying the global optima, if it is located outside the region, covered by the history of search. The trace of memory-based noise handling strategy is also found in [199].

3.5. Fitness inheritence

Fitness inheritance methodology, realized in noisy genetic algorithm in [44], aims at estimating the fitness of an offspring based on the fitness measures of its pair of parents. Let $\overline{f}(\overrightarrow{X_j})$ and $\overline{f}(\overrightarrow{X_k})$ be the mean fitness estimates of the parents, $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$, of a given trial solution $\overrightarrow{X}$. The fitness variances of $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$ are denoted by $\sigma(f(\overrightarrow{X_j}))$ and $\sigma(f(\overrightarrow{X_k}))$ respectively. Instead of resampling the offspring $\overrightarrow{X}$, it is first evaluated only once. Let the fitness of $\overrightarrow{X}$ after single evaluation be $f(\overrightarrow{X})$. The evaluated fitness $f(\overrightarrow{X})$ is accepted as the fitness estimate of $\overrightarrow{X}$ if it satisfies the following statistical comparison, given by

$$\mu - 3\delta < f(\overrightarrow{X}) < \mu + 3\delta \tag{73}$$

where,

$$\mu = \frac{\overline{f}(\overrightarrow{X_j}) + \overline{f}(\overrightarrow{X_k})}{2} \tag{74}$$

and

$$\delta = \frac{\sigma(f(\overrightarrow{X_j})) + \sigma(f(\overrightarrow{X_k}))}{2} \tag{75}$$

Hence the approach eliminates the computational overhead associated with periodic evaluations of $\overrightarrow{X}$. However, if $f(\overrightarrow{X})$ fails to satisfy the condition, the need for sampling $\overrightarrow{X}$ becomes necessary to reduce the adverse effect of noise infiltration in its objective function value. Under such scenario, $\overrightarrow{X}$ is sampled for a pre-specified number of times. The resulting fitness samples of $\overrightarrow{X}$ are then averaged to return its mean fitness estimate $\overline{f}(\overrightarrow{X})$.

Although the proposed strategy ensures reduction in the computational complexity by avoiding unnecessary re-evaluations of a trial solution if it satisfies the statistical test, it may fail to precisely locate the global optima in the objective surface contaminated with noise of large variance. It is evident from (75) that larger the average value δ of the fitness variance of parent solutions (indicating a large scale adverse effect of noise), greater is the possibility of accepting the single fitness estimate (without resampling) of the offspring trial solution. Apparently, it directs the search towards noise-prone area in the search space.

4. Noisy optimization using implicit averaging

An alternative approach to explicit averaging is *implicit averaging*, which aims at enlarging a population size of an EA to reduce the contamination effect of noise in a trial solution. It stems from the supposition that an EA, with its population-based meta-heuristic explorative capability, usually encounters with the same trial solutions often during searching in the parameter space over evolutionary generations. Every time it comes across the same trial solution through its evolutionary search dynamics, it evaluates its fitness, effectively re-evaluating the given trial solution periodically [7,85]. The strategy is thus named as *internal sampling* or *averaging*. Apparently, to reduce the adverse effect of infiltration of noise in the fitness landscape using implicit or internal sampling, the size of the population needs to be large enough [10,56,57,84].

A comparative analysis of the relative robustness of traditional population-based over the point-based search strategies is undertaken in [139,140]. In each category, two members are considered. The evolutionary strategy (ES) [27] and genetic algorithm (GA) [129] are considered as members of population-based optimization methods while pattern search (PS) [87] and threshold acceptance (TA) [64] are considered as the members of the point-based search method. The performances of the algorithms of both categories remain comparable without employing sampling policy. However, when each solution is re-evaluated with fixed sample size (say, 10), the population-based methods in noisy environment clearly outperforms the point-based search algorithms. The performance of point-based methods is affected by a *mediocre* solution with illusive good fitness, especially in complex fitness terrain. Due to less explorative capability, these algorithms waste significant computational resources in searching in the local neighborhoods of such misleading solutions. Contrarily, the inherent exploration capability and evolutionary dynamic of population-based methods help to identify the promising area near the global optima,

even when noise is a predominant factor.

The effect of population sizing on the efficiency of (1, λ)-ES to overcome the noise infiltration is discussed in [7]. The conventional mutation strategy of (1, λ)-ES in a D-dimensional search space is described by

$$\vec{Y} = \vec{X} + \vec{Z} \tag{76}$$

where $\vec{X}$ is the parent trial solution, $\vec{Y}$ is the resulting offspring solution and $\vec{Z}$ is the mutation vector. Considering isotropic normal mutation scheme, the components of $\vec{Z}$ are normally distributed with mean 0 and variance s^2. The standard deviation s of the components of $\vec{Z}$ are referred to as the *mutation strength*. The offspring $\vec{Y}$ replaces the respective parent $\vec{X}$ if is better than $\vec{X}$ in terms of fitness.

In the noisy fitness landscape, the measured fitness of $\vec{X}$, is represented as

$$f_{noisy}(\vec{X}) = f(\vec{X}) + \eta \tag{77}$$

where the injected noise η is assumed normally distributed noise with mean 0 and variance σ_ε^2, often referred to as the *noise strength*. Let R_x and R_y be the respective distances of $\vec{X}$ and $\vec{Y}$ from the true global optimum $\hat{\vec{X}}$ in a D-dimensional spherical objective surface. The progress rate φ is then defined as

$$\varphi = E(R_x - R_y). \tag{78}$$

Further, it is shown in [7,9,11] that the mutation vector $\vec{Z}$ can be decomposed into two vector components (i) $\vec{Z}_A = \{z_1, 0, \dots, 0\}$ parallel to $\vec{X} - \hat{\vec{X}}$, and (ii) $\vec{Z}_B = \{0, z_2, \dots, z_D\}$ perpendicular to the plane of $\vec{Z}_A$, as shown in Fig. 10. It is evident from Fig. 10 that

$$R_y^2 = (R_x - z_1)^2 + \|\vec{Z}_B\|^2. \tag{79}$$

As the components of $\vec{Z}$ follow zero mean normal distribution with variance s^2, (79) can be approximated by

$$R_y^2 \cong R_x^2 - 2R_x z_1 + Ds^2 \tag{80}$$

for $D\to\infty$. The normalized progress rate is then obtained by

$$\hat{\varphi} = c_{1,\lambda}\hat{s}\frac{1}{\sqrt{1 + (\hat{\sigma}_\varepsilon/\hat{s})^2}} - \frac{\hat{s}^2}{2} \tag{81}$$

where $c_{1,\lambda}$ is the progress coefficient. The normalized mutation and noise strengths are respectively given by [28]

$$\hat{s} = s\frac{D}{R_x} \text{ and } \hat{\sigma}_\varepsilon = \frac{\sigma_\varepsilon}{\left|\frac{df(\vec{X})}{dR_x}\right|}\frac{D}{R_x}. \tag{82}$$

Based on the *evolutionary progress principle* [21] and from (81), it is apparent that the effective evolutionary progress is a function of two opposing forces,

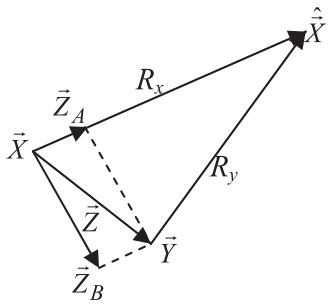

Fig. 10. Decomposition of the mutation vector into two perpendicular components.

$$Evolutionary\ \ Gain = c_{1,\lambda}\hat{s}\frac{1}{\sqrt{1 + (\hat{\sigma}_\varepsilon/\hat{s})^2}} \tag{83}$$

and

$$Evolutionary\ \ Loss = \frac{\hat{s}^2}{2}. \tag{84}$$

The evolutionary loss part is thus independent of the normalized noise strength $\hat{\sigma}_\varepsilon^2$. To obtain an overall evolutionary progress with constant $\hat{\sigma}_\varepsilon^2$ and $\hat{s}$, the evolutionary gain part needs to be effectively enhanced by increasing the progress coefficient $c_{1,\lambda}$. It is noteworthy that $c_{1,\lambda}$ is a function of the offspring population size λ, given by

$$c_{1,\lambda} \sim \sqrt{2\ln\lambda}. \tag{85}$$

It indirectly implies that larger the population size λ, higher is the value of $c_{1,\lambda}$ and also the evolutionary gain part, leading to a net effective evolutionary progress towards the true global optimum in the noisy fitness landscape.

Moreover, it is revealed in [10] that a significantly large value is obtained for the net evolutionary progress in noisy environment in case of (μ, λ)-ES by increasing the parent population size μ. It is due to the functional dependence of $c_{\mu,\lambda}$ on the population variance $\delta=D^2/s^2$ and the noise level $\vartheta=\sigma_\varepsilon/s$, represented by

$$c_{\mu,\lambda} = \frac{1+\delta}{\sqrt{1+\delta+\vartheta^2}} e_{\mu,\lambda}^{1,0} \tag{86}$$

Here, $e_{\mu,\lambda}^{1,0}$ is an instance of the general progress coefficients [13]. For (1, λ)-ES, μ=1 and hence δ=0. It is also pointed out in [13] that the effective progress gain of (μ, λ)-ES is almost tripled than that of (1, λ)-ES for a given value of λ and fixed noise level ϑ. For example, $c_{1,100}$=0.31 while $c_{22,100}$=0.8 for λ=100 and ϑ=8.

The effect of resampling reveals to be less efficient in handling noise as compared to large population size in [33]. The restart mechanism, realized with the *adapted maximum-likelihood Gaussian model* of estimation-of-distributed algorithm (EDA) [120,147] increases the population size. This enables the algorithm to achieve significant robustness against noise in high-dimensional search space. In [190], the population size is dynamically adapted during the run-time of a multi-objective EA based on the characteristics of the objective surface and the desired population diversity.

The beneficial effect of population sizing over sampling strategy is also studied in [74,82]. Let each trial solution $\vec{X}$ is assigned with a sample size n, resulting in the mean fitness estimate $\overline{f}(\vec{X})$ and the fitness variance $\sigma(f(\vec{X}))$. It is shown in [74] that the random variable S, resulting from averaging the mean fitness estimates of r randomly selected trial solution (or *structures*) from a *hyper-plane* H, has a mean μ_S and variance σ_S^2, given by

$$\mu_s = \langle \overline{f}(\vec{X}_j)\rangle_H \tag{87}$$

and

$$\sigma_s^2 = \frac{1}{r}\sigma^2 + \frac{1}{rn}\langle\sigma^2(f(\vec{X}_j))\rangle_H \tag{88}$$

where

$$\sigma^2 = \langle\eta^2(\vec{X}_j)\rangle_H \text{ and } \eta(\vec{X}_j) = \overline{f}(\vec{X}_j) - \mu_s \tag{89}$$

Evidently, the total number of samples used to determine the performance of the hyper-plane H is $r\times n$. It is apparent from (88) that for a fixed value of $r\times n$, the performance of the hyper-plane can be enhanced by reducing n and increasing r, which is proportional to the population size NP. The effect of sampling in modulating σ_S^2 is identified by considering r constant, giving

$$\frac{\delta\sigma_s^2}{\delta n} = \frac{1}{r}\frac{\delta\langle\sigma_n^2(f(\overrightarrow{X_j}))\rangle_H}{\delta n} \tag{90}$$

where

$$\sigma_n^2(f(\overrightarrow{X})) = \frac{\sigma^2(f(\overrightarrow{X}))}{n}. \tag{91}$$

The expression (90) indicates that with an increase in the sample size n per trial solution, the performance estimate of the hyper-plane is increased only by a factor of $1/r$ with respect to the average accuracy of the individual trial solutions in the hyper-plane.

An analysis based on the trade-off between the sample size and the population size is also undertaken in [74] by setting the total time taken by a genetic algorithm as

$$T = (\alpha + \beta n)G \times NP \tag{92}$$

with $n \times NP$ as constant. Here G represents the number of evolutionary generations, α denotes the computational cost per trial solution (allocated by genetic algorithm) and β symbolizes the computational cost incurred by evaluating each fitness sample of a trial solution. In the present context, it is evident that the number of generations G that can be reached within a constant time T decreases with an increase in the population size NP, leading to the faster convergence. A case study in an image processing application also substantiates that the implicit averaging occasionally outperforms the explicit averaging to improve the performance of an EA in presence of noise.

Although the implicit averaging strategy increases the population diversity (by effectively exploring the entire search space using a large population) and hence increases the robustness against noise, it suffers from three shortcomings. First, it extensively increases the computational overhead by evaluating the objective function values of all members of a large population over the search landscape. Second, it leads to redundancy by evaluating the same solution over evolutionary generations. Lastly, the convergence of an EA utilizing implicit averaging also significantly depends on the selection process. If the selection operation is deceived by noise, it will reduce the selection pressure and deteriorates the performance of the algorithm [17].

5. Improved evolutionary search strategies for noisy optimization

There exists extensive literature on efficiently guiding the evolutionary search towards optimal region in the noise-induced fitness landscape. Two major challenges for devising noise-tolerant evolutionary search strategies include (i) efficiently examining the local neighborhood of a trial solution using robust exploitation operations, and (ii) capability to explore the search space to avoid premature convergence [107]. The theoretical aspects and comparative analysis of a few evolutionary strategies in noisy environment are summarized in [6,29,95]. We present some eminent robust search dynamics proposed in the literature to rescue the population members from the illusory consequences of noise in the fitness landscape [18,19].

5.1. Robust features of multi-objective evolutionary algorithm

Inspired by the role of the momentum term in back-propagation in neural networks, Goh and Tan [77,78,79] adopted an *experiential-learning-directed perturbation* (ELDP) strategy to govern the candidate movement toward the direction of fitness improvement on the basis of the information acquired from the last few generations for faster convergence. ELDP strategy differs from conventional crossover mechanism in two aspects, including (i) the ordered change in the trial solution (or chromosome) instead of being random and (ii) the variation being performed either in genotype or phenotype space. Let the decision vector be represented by $\vec{\zeta}$ and $\vec{\rho}$ in the genotype and the phenotype spaces respectively while G denoting the evolutionary generation index. According to the ELDP strategy, the j-th decision variable (of the trial solution) in the phenotype is modified by

$$\overline{\Delta}\rho_j(G) = \Delta\rho_j(G) + \alpha \times \overline{\Delta}\rho_j(G-1) \tag{93}$$

where Δ and $\overline{\Delta}$ respectively signify the changes incurred in the phenotype space of the given trial solution through genetic operators (such as crossover) in the current generation and over prior generations (considering the effect of momentum). α represents the learning rate. Apparently, the variation is increased (or decreased) in the direction of possible changes in the phenotype space, if $\Delta\rho_j(G)$ and $\overline{\Delta}\rho_j(G-1)$ are same (or opposite) in polarity, particularly in the earlier (or convergence) phase to perform exploration (or local fine tuning) efficiently. Accordingly, the new phenotype representation of the trial solution is obtained by

$$\rho_j(G+1) = \begin{cases} \rho_j(G) + \overline{\Delta}\rho_j(G), & \text{if } \Delta^{\min} < \alpha \times \overline{\Delta}\rho_j(G) < \Delta^{\max} \\ f(BF(\zeta_j(G) + \Delta\zeta_j)), & \text{otherwise} \end{cases} \tag{94}$$

Here, $BF(.)$ represents the bit flip-flop mutation for the j-th decision variable. Limiting the extent of directed perturbation by the bounds $[\Delta^{\min}, \Delta^{\max}]$, the strategy simultaneously ensures exploration in new directions in the search space, induced by bit flip-flop operator. It is especially beneficial to get rid of any possible outlier or to expedite the shortly stalled evolutionary search process. Hence, the objective of ELDP is two folds, (i) controlling the directed perturbation in the phenotype space and (ii) enhancing the population diversity by bit flip-flop mutation in genotype space.

Moreover, the trade-off between the explorative and exploitative capabilities of the MOO here are balanced by the proposed *gene adaptation selection strategy* (GASS). It is concerned with modulating the population diversity to prevent the members from (i) premature convergence and (ii) participating in a completely random search in noisy environment. The *convergence model* of GASS re-defines the search space by increasing the search range to prevent premature convergence. The *divergence model*, on the other hand, decreases the possibility of random change in the decision variables by reducing the search range. Evidently, convergence model is preferred in the earlier exploration phase while prioritizing the divergence model when the archive has reached close to its full capacity. Finally a *possibility and necessity* measure induced *possibilistic archiving* method is used to resolve the uncertainty in including a solution in the non-dominated set.

5.2. Mutation strength

Coined by Rechenberg [163], significant focuses are made by researchers on adapting the mutation strength of an EA [8]. The effect of rescaled mutation of $(1, \lambda)$-ES in a noisy sphere objective surface is studied in [23]. The normalized evolutionary progress rate $\hat{\varphi}$, comprising two components: evolutionary gain and evolutionary loss, has already been produced in (81) for a traditional $(1, \lambda)$-ES.

If the normalized noise strength $\hat{\sigma}_\varepsilon^2$ is constant, the evolutionary gain part can be increased by two alternative approaches. The first approach deals with increasing the offspring population size λ, as the progress coefficient $c_{1,\lambda} \sim (2\ln\lambda)^{1/2}$. The second alternative aims at reducing noise strength $\hat{\sigma}_\varepsilon^2$ by resampling each individual. However, both the approaches invariably increase the computational complexity, associated with fitness evaluations.

Motivated by this observation, an interesting strategy is proposed in [23] to reduce the evolutionary loss part by rescaling the mutation strategy, as given by

$$\vec{Y} = \overrightarrow{X} + \frac{1}{k}\overrightarrow{Z},\ k > 1 \tag{95}$$

with $\overrightarrow{X}$, $\vec{Y}$ and $\overrightarrow{Z}$ symbolizing the parent, offspring and mutation

vectors respectively.

Due to the rescaled mutation with scale factor k, the normalized mutation strength $\hat{s}$ is also reduced by a factor of k. If $\hat{\sigma}_\varepsilon << \hat{s}$, the evolutionary gain part is a linear function of $\hat{s}$ while the evolutionary loss part is a quadratic function of $\hat{s}$. Hence rescaling mutation with the scale factor k, reduces the loss part more than the gain part, leading to a net effective evolutionary progress. Under rescaling operation, the normalized progress rate is reformulated as

$$\hat{\varphi} = \frac{1}{k}\left[c_{1,\lambda}\hat{s}\frac{1}{\sqrt{1+(\hat{\sigma}_\varepsilon/\hat{s})^2}} - \frac{\hat{s}^2}{2k}\right] \tag{96}$$

In [12], the mutation strength of $(\mu/\mu, \lambda)$-ES is adapted with the *cumulated progress vector*, which holds a record of the past steps taken by the evolutionary strategy. It is observed that the adapted mutation strength becomes too large (or small) in the presence (or absence) of noise in the fitness landscape.

The influential effect of mutation strategy in *bio-geography based optimization* (BBO) [180,181] is studied in [121] for improving the probability of identifying true global optima in a noisy objective surface. The Markov-chain formulation of the migration probability is used to indicate that the probability of transition of a trial solution to the global optimum gradually deteriorates with its increasing (noise-driven) fitness deviation from the true value. This inefficacy of migration scheme to handle the jeopardizing effect of noise, however, can be alleviated by mutation strategy with moderately high mutation probability.

5.3. Data mining approach

In [51], a *Bayesian frequent data mining* [76] induced single-point crossover is applied in noisy multi-objective evolutionary algorithm. It provides a *direct search* strategy to assist in the discovery of the optimal regions in the objective surfaces. It also fine-tunes the local search in the identified potential zone. To accomplish the data mining driven directed search, first the ranges of the objectives and that of each decision variable are divided into a pre-specified number of intervals. Next for each individual, the Bayesian conditional probability is calculated which captures the probability that the given individual is non-dominated given that its decision variable belongs to a specific interval. This is repeated for all decision variables in a D-dimensional search space. This Bayesian conditional probability is then utilized to identify the possible range of each decision variable, which is presumed to give a non-dominated solution. These D *orthotopes* (each for one decision variable) now help in identifying the optimal region in the multiple fitness landscapes, denoted by R^{opt}. A R^{opt} is uniquely identified by its D-dimensional geometric centre $\overrightarrow{C}^{opt}(G)$. It is mention worthy that R^{opt} varies over evolutionary generations. It may happen that R^{opt} is incorrectly detected in the presence of noise or it may be far away from the true global optima. To hinder the erratic movement of R^{opt} in the parameter space under the influence of contaminating noise, R^{opt} of a specific generation G is combined with R^{opt} of previous generations using a *moving average* formulation. This new optimal region in the objective surfaces is symbolized by $R^{opt,MA}$. It is obtained by setting

$$\overrightarrow{C}^{MA}(G) = \frac{(G-1)\times\overrightarrow{C}^{MA}(G-1) + \overrightarrow{C}^{opt}(G)}{G} \tag{97}$$

Here, $\overrightarrow{C}^{MA}(G)$ and $\overrightarrow{C}^{opt}(G)$ respectively represent the geometric centers of $R^{opt,MA}(G)$ and $R^{opt}(G)$ (obtained by D orthotopes of current generation G without considering the effects from previous generations). The directed search mechanism now makes a trial solution $\overrightarrow{X}(G)$ biased towards the potential zones in the objective surfaces by setting

$$\overrightarrow{X}(G+1) = \overrightarrow{X}(G) + (\overrightarrow{C}^{MA}(G) - \overrightarrow{X}(G)) \tag{98}$$

The local confinement of search process in the promising zones is counterbalanced by the *extended exploration* phase of the algorithm to ensure population diversity.

5.4. Memetic approach

The uncertainty handling policy in [132], realized with *differential evolution* (DE) [59,151,153,187] in multimodal high-dimensional search space, encompasses three main steps, including (i) *controlled randomization of parameters* to enhance robustness of DE against noise, (ii) *scale factor local search* for generation of promising offspring vectors and (iii) a statistical comparison for judicious selection of trial solutions over generations. The optimization proficiency of DE is determined by its efficacy of generating potential trial solutions using evolutionary dynamics, like mutation and crossover. The performance of DE relies on the appropriate settings of the control parameters, including the scale factor, F and the crossover ratio, CR. The deterministic behavior of traditional DE, induced by constant values of control parameters, is less competent to handle uncertainties due to presence of noise in the multimodal fitness landscape. Contrarily, a completely randomized update of the control parameters may result in a haphazard movement of trial solutions, slowing down the algorithmic convergence. To circumvent the problem, a controlled randomization of control parameters [40] is utilized here to ensure a balance between explorative and exploitative requirements.

However, the scale factor F^{best}, used for generation of an offspring corresponding to the best parent solution $\overrightarrow{X}^{best}$, can be alternatively identified by local search mechanisms, including *golden selection search* (GSS) [103] and *hill climbing* (HC) [166]. These strategies help to select the optimal value of F^{best} by searching locally in the prescribed bound of the scale factor such that the search direction, indicated by $\overrightarrow{X}_{r1} + F^{best}(\overrightarrow{X}_{r2} - \overrightarrow{X}_{r3})$ provides the best fitness of an offspring solution for a given set of $\overrightarrow{X}$, $\overrightarrow{X}_{r1}$, $\overrightarrow{X}_{r2}$ and $\overrightarrow{X}_{r3}$. The best trial solution $\overrightarrow{X}^{best}$ undergoes through the mutation and crossover phases with control parameters being determined using GSS or HC or controlled randomization based on the respective probabilities of p_g, p_h and $(1-p_g-p_h)$. The local search strategies, however, are not applied for generation of offspring vectors corresponding to all parent solutions apart from $\overrightarrow{X}^{best}$.

5.5. Adaptive walk

The behavior of adaptive walk in presence of noise is studied in [112]. The *agents* (or trial solutions) remain successful in identifying the local basins of attraction successfully during *noiseless walk*. Little presence of noise, however, is revealed beneficial for escaping the local optima during the adaptive walk of trial solutions. It is also exposed that, in presence of noise, constant or exponentially decreasing with fitness, the noisy adaptive walks first help the agents to wander through the fitness landscape until they discover a specific point in the objective surface with relatively high fitness, referred to as *attractor*. They are thus clustered around the attractor. Eventually they make transition from random walk to hill climbing the fitness landscape to reach the local peaks (for maximization problem). The rambling movement of agents is widespread in the objective surface with increasing noise variance.

5.6. Chaotic jump

In [124], a *chaotic jump* (CJ) strategy is introduced to help the trial solutions in escaping local optima and false basins of attraction in the noisy fitness landscape. If a trial solution is obtained with no fitness

improvement for a specific duration, called *maximum_stagnation_interval* (*MSI*), the solution jumps towards a new position in the parameter space. The amount of chaotic jump is determined by the chaotic sequence following the *chaotic logistic map* [67]. The strategy has been realized with four variants of *particle swarm optimization* (PSO), including global best PSO [53], local best PSO with ring topology [4], fully informed PSO and bare-bone PSO (BBPSO) [102]. If the current position of a particle in PSO remains unsuccessful in replacing its respective *personal best position* (p^{best}) for a period greater than *MSI*, the chaotic jump is performed by the particle. It thus provides an explorative capability to handle noise. Among the four comparative variants, BBPSO-CJ is shown to outperform other competitors in the noisy environment due to its efficiency in sampling the search space according to a normal distribution.

5.7. Local model-based search

In [45], the noise in evolutionary multi-objective optimization is dealt with by dividing the entire D-dimensional search space into S (a pre-specified number) spheres. Each sphere is assigned with a pair of centroid and radius, which are adapted over evolutionary generations. All the spheres together define the local models of the entire system in D-dimensional parameter space. The initial radius (at evolutionary generation G=0) $r_i(G=0)=r$ of each sphere is determined by satisfying $r \leq R/\sqrt[D]{S}$ for i=[1, S]. Here R is the smallest possible radius of a sphere covering the entire D-dimensional search space, given by

$$R = \sqrt{\sum_{j=1}^{D} (x_j^{\max} - x_j^{\min})^2}/2 \tag{99}$$

where $x_j^{\min}$ and $x_j^{\max}$ represent the minimum and maximum bound in the j-th dimension for j=[1, D]. To ensure the exploration in earlier evolutionary phase, the centroids of the spheres are determined such that the initialized spheres are non-overlapping. Let $\vec{C}_i(G)$ be the D-dimensional centroid of the i-th sphere at generation G

$$|\vec{C}_i(0) - \vec{C}_j(0)| \geq 2r \quad \text{for } i, j = [1, S]. \tag{100}$$

A number of candidate solutions are then initialized in each sphere, which subsequently undergoes one evolution of the algorithm. After that, the non-dominated trial solutions in each sphere are identified. The centroids of the spheres are then updated based on the location information of their non-dominated members. The local best centroid $\vec{C}_i^{lb}(G)$ of the i-th sphere is determined by averaging the position of its non-dominated solutions. In other words,

$$\vec{C}_i^{lb}(G) = \sqrt{\sum_{j=1}^{m_i} \vec{X}_{i,j}}/m_i \tag{101}$$

where m_i is the number of non-dominated solutions in the i-th sphere. Here $\vec{X}_{i,j}$ denotes the j-th non-dominated solution belonging to the i-th sphere. The centroid of the archive is considered as the global best centroid, symbolized by $\vec{C}_i^{gb}(G)$. The new centroids of the spheres are next obtained by following the traditional PSO dynamics.

$$\vec{v}_i(G+1) = \chi\Big[\omega\vec{v}_i(G) + C_1\phi_1(\vec{C}_i^{lb}(G) - \vec{C}_i(G)) + C_2\phi_2(\vec{C}_i^{gb}(G) - \vec{C}_i(G))\Big] \tag{102}$$

$$\vec{C}_i(G+1) = \vec{C}_i(G) + \vec{v}_i(G+1) \tag{103}$$

Here $\vec{v}_i(G)$ is the velocity of the i-th centroid, which is initialized with $\vec{v}_i(0)$=0. Hence the overall direction of improvement, given by

$$\vec{D}(G) = \sum_{i=1}^{S} (\vec{C}_i(G+1) - \vec{C}_i(G)) \tag{104}$$

eventually guide the population members (within the local spherical models) towards the global optima. The composite opinion of all spheres in the evaluation of the net direction of improvement is expected to nullify the deceptive effect of noise and thus helps to discover quality solutions over evolutionary generations.

Apart from the above-mentioned approaches of noise handling using evolutionary search dynamics, the works proposed in [20,72,90,105,122,152,183] need special mentioning.

6. Selection in noisy optimization

In noisy evolutionary optimization problem, the robustness of selection operation against noise [130] is influenced by two important factors, including (i) the judicious selection of trial solutions for being resampled to reduce the noise-induced uncertainty, and (ii) the promotion of truly quality solutions over evolutionary generations [69,115]. The first category presumes that the information content of a quality solution will effectively guide the evolutionary search over evolutionary generations in presence of uncertainty. The second category of selection process is concerned with identifying the better solution form a pair (for single objective optimization) or a non-dominated pool (in case of multi-objective optimization) of solutions to pass onto the next generation. Researchers are taking keen interest to develop robust selection strategies to enhance the reliability in selecting quality solutions surmounting the jeopardizing effect of noise. Some well-known selection strategies in presence of noise are presented next.

6.1. Kalman extension of genetic algorithm

In [188], a *Kalman-extended genetic algorithm* is proposed which utilizes a specified fraction of the maximum number of evolutionary generations to generate new offsprings, while devoting the remaining evolutionary cycles to the re-evaluation of an existing population member. The devised mechanism aims at identifying the population member to be selected for re-evaluation. The main objective of the selection criterion is not to waste significant computational time by re-evaluating trial solutions with poor fitness estimate, or with very low uncertainty. Useful knowledge about the effect of noise in the promising zones of the fitness landscape can be effectively captured by the re-evaluations of quality solutions with more uncertainty in their fitness estimates. The strategy thus selects an existing candidate with the highest noise-induced uncertainty from the set of members having fitness greater than a threshold (for maximization problem). The threshold is set to the difference between the population mean fitness and the standard deviation of the population members' fitness estimates. The noise introducing uncertainty in fitness estimates of trial solutions here is handled as *observational noise* of Kalman formulation, which is eventually reduced over generations.

6.2. Uncertainty quantification and treatment for online optimization

A novel strategy is proposed in [86,108] for handling noise-induced uncertainty in evolutionary optimization algorithms which employ *rank-based selection*. The proposed strategy is realized with *covariance matrix adaptation-evolutionary strategy* (CMA-ES) and named as *uncertainty handling* (*UH*) *CMA-ES*. It is then applied for the online optimization of feedback controllers in a combustor test rig. The proposed methodology is partitioned into two phases. The first phase deals with the *quantification of uncertainty*. Here a randomly selected set of solutions are first re-evaluated. The re-evaluation may result in the change in the ranking of the population members. The measured rank changes are then use for capturing the uncertainty in the current generation population. The next phase, called *uncertainty treatment*, is concerned with following two aspects.

1. It increases (or decreases) the evaluation/measuring time of the performance of the controller based on a larger (or smaller) average estimate of uncertainty, as quantified in the previous phase.
2. It increases the population variance to (i) enhance signal-to-noise ratio (SNR), (ii) help trial solutions to escape local optima with low SNR and (iii) prevent premature convergence of CMA-ES and assist in exploring the search space.

6.3. Rolling tide selection

In [71], a *rolling tide* characteristic is introduced into an elitist multi-objective optimization to avoid deception by noise while selecting optimal solutions and identifying the solution for resampling. Each population member of the algorithm is assigned with a *single-track dominator* from trial solutions of the non-dominated elite archive A. Initially, all the objectives of each population member are evaluated only for once. Whenever a new offspring solution $\overrightarrow{X}$ is created, it is inserted into A if it is not dominated by any member of A. The members of A dominated by $\overrightarrow{X}$ are excluded from A and are put into the dominated pool with $\overrightarrow{X}$, recorded as their single track dominator. Otherwise, a solution from A, which dominates $\overrightarrow{X}$, is referred to as the single-track dominator of $\overrightarrow{X}$. After updating the archive, the trial solution $\overrightarrow{X}'$ is identified from A with the minimum number of re-evaluations of objective functions. The set of solutions $dom(\overrightarrow{X}')$, assigned with $\overrightarrow{X}'$ as their single-track dominator, is also extracted from the dominated pool. $\overrightarrow{X}'$ is then sampled for one more time. The additional resampling of $\overrightarrow{X}'$ may change its dominance relationship with members of $dom(\overrightarrow{X}')$. If $\overrightarrow{X}'$ does not further dominate members of $dom(\overrightarrow{X}')$, $\overrightarrow{X}'$ is removed from A and its single track dominator is identified from $A \cup dom(\overrightarrow{X}')$. Moreover, the strategy also focuses either on adaptively increasing the rate of resampling if the elite archive gets stalled (or oscillatory due to few resampling), or the strategy concentrates on evaluations of new offsprings if the archive is progressing with evolutionary generations respectively.

6.4. Dominance dependent lifetime

In [42], three noise-tolerant extensions are incorporated in traditional multi-objective optimization to solve noisy combustion process. First, each population member is assigned with a *dominance-dependent lifetime*, which varies inversely with the number of solutions it dominates. Thus, the strategy defends the overall population against the impact of the misleading fitness of unreliable solutions (Fig. 11).

Conversely, the survival of quality solutions is ensured by the second extension. It is concerned with reevaluating the archived solutions with expired lifetime and adding them to the current population. While the key role of dominance dependent lifetime is to get rid of outliers with deceptive good fitness, the re-evaluations of quality solutions with expired lifetime (i.e., with a great potential to dominate other contenders) may help them to re-enter the archive. Evidently, a true quality solution always returns good fitness after re-evaluations but it is unexpected that an outlier, previously assessed as a good member, will again possess the good fitness estimate. Hence, the re-evaluation assists in preserving quality candidates in the archive while eradicating the outliers. Once the true noisy solutions are removed from the archive, a large fraction of population, previously dominated by the noisy ones (or outliers), become eventually non-dominated. However, this set of solutions previously was not accommodated in the archive due to the presence of their illusively dominating noisy solutions (or outliers). This may lead to the loss of information in noisy environments. Finally, this loss is hindered by appending all these non-dominated solutions (previously dominated by noisy solutions or outliers) with nonexpired lifetime, to the archive.

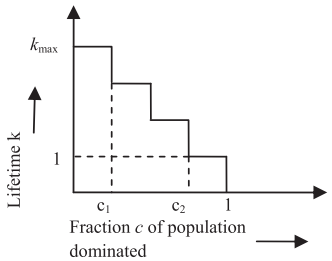

Fig. 11. Transformation of lifetime of a solution based on the fraction of entire population it dominates.

6.5. Selection using statistical comparator

The fitness-based Pareto ranking adopted for placement of the members in the Pareto fronts (during non-dominated sorting of multi-objective optimization) may not be always correct due noise infiltration. Let $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$ be two population members with respective Pareto ranks $R(\overrightarrow{X_j})=1$ and $R(\overrightarrow{X_k})>1$. Evidently, $\overrightarrow{X_k}$ is dominated by $\overrightarrow{X_j}$. To give the relatively worse-looking member $\overrightarrow{X_k}$ a chance to occupy the approximate Pareto front along with $\overrightarrow{X_j}$, Goldberg introduced the following strategy [14]. The strategy checks whether the difference between the sample means of $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$ is less than the scaled sum of their variances, i.e.,

$$|\overline{f}_i(\overrightarrow{X_j}) - \overline{f}_i(\overrightarrow{X_k})| < \alpha\sqrt{(\sigma^2(f_i(\overrightarrow{X_j})) + \sigma^2(f_i(\overrightarrow{X_k})))/2} \tag{105}$$

where, $\overline{f}_i(\overrightarrow{X_j})$ denotes the mean value of the samples of i-th objective $f_i(\overrightarrow{X_j})$ for i=[1, N]. α is a scale factor, called *neighborhood restriction parameter*.

In [154], Goldberg's method is amended as follows:

1. First, the mean $\overline{f}_i(\overrightarrow{X_j})$ is replaced by expectation $E(f_i(\overrightarrow{X_j}))$ of the fitness samples for individual population member for the i-th objective with i=[1, N], as it provides a better measure of the statistical fitness than the mean estimate.
2. The sample size for the two members being different, as in [154], the variance is substituted by the variance divided by sample size, using Welch's t-test criterion [111].

Incorporation of above mentioned points, yields a new criterion for testing Pareto co-ranking of two members of the population.

$$|E(f_i(\overrightarrow{X_j})) - E(f_i(\overrightarrow{X_k}))| < \alpha\sqrt{\left(\frac{\sigma^2(f_i(\overrightarrow{X_j}))}{n(\overrightarrow{X_j})} + \frac{\sigma^2(f_i(\overrightarrow{X_k}))}{n(\overrightarrow{X_k})}\right)} \tag{106}$$

In [156], the centroidal fitness estimates of individual candidates are compared to satisfy a more viable statistical comparator is used in [156,157]. Here, the Welch criterion is replaced by pooled variance with an aim to capture the uncertainty of Pareto ranking of population members more efficiently. It is given by

$$|\widetilde{f}_{i,c}(\overrightarrow{X_j}) - \widetilde{f}_{i,c}(\overrightarrow{X_k})| < \alpha \times \sqrt{PV_i(\overrightarrow{X_j}, \overrightarrow{X_k})}, \quad \text{for } i = [1, N] \tag{107}$$

where the pooled variance [1] is given by

$$PV_i(\overrightarrow{X_j}, \overrightarrow{X_k}) = \frac{(n(\overrightarrow{X_j}) - 1) \times \sigma^2(f_i(\overrightarrow{X_j})) + (n(\overrightarrow{X_k}) - 1) \times \sigma^2(f_i(\overrightarrow{X_k}))}{n(\overrightarrow{X_j}) + n(\overrightarrow{X_k}) - 2}. \tag{108}$$

To keep the slightly worse solutions a chance to enter the Pareto optimal front during the early exploration phase of the evolutionary algorithm and restrict their entry gradually, a model is employed to design α that captures the above necessity automatically.

$$\alpha = C[1 - \exp(-\sigma_N^2/G)] \tag{109}$$

where $\sigma_N{}^2$ denotes the noise variance. When generation G is a small positive integer, $\alpha \to C$ (≈10, selected empirically) while for a large G (theoretically infinity) α approaches zero.

A hypothesis test is performed for the competitive selection in [182] in multi-objective evolutionary optimization. A student t-distribution based statistical significant test is undertaken to identify the truly dominant solution among a pair of contenders. However, if the test is inconclusive, the solution with less uncertainty is preferred. Apart from t-test, two additional tests, including *Wilcoxon* and *median* tests are also performed to select quality neighbors of a trial solution in *Tabu Search* (TS) [55]. In [58], an offspring vector is permitted an entry in the next generation if its fitness is better than the fitness of its respective parent by a threshold. The threshold here has been made proportional to the noise variance. The study in [65] addressed the concepts of stochastic and significant dominance to effectively discriminate among competing solutions in noisy fitness landscape. Among other approaches, considering statistical test based selection of quality solutions, the works proposed in [32,127,128,148,165] need special mentioning.

6.6. Probabilistic dominance

In a multi-objective optimization problem (where all N objectives are to be jointly minimized), a trial solution $\overrightarrow{X_j}$ is said to dominate $\overrightarrow{X_k}$, denoted by $\overrightarrow{X_j} \prec \overrightarrow{X_k}$, if the following conditions are jointly satisfied.

$$f_i(\overrightarrow{X_j}) \le f_i(\overrightarrow{X_k}) \text{ for } i = [1, N]. \tag{110}$$

$$f_l(\overrightarrow{X_j}) < f_l(\overrightarrow{X_k}) \text{ for at least one } l \in [1, N]. \tag{111}$$

The strict inequality conditions of dominance of $\overrightarrow{X_j}$ over $\overrightarrow{X_k}$ cannot correctly validate the dominance criteria when $f_i(\overrightarrow{X_j})$ and $f_i(\overrightarrow{X_k})$ are both contaminated with noise for i=[1, N]. Deterministic dismissal of apparently inferior trial solutions from optimal Pareto front can lead to the loss of a potentially good solution in the presence of noise. To overcome this, stochastic dominance criteria are adopted in [70,191,197], to identify the degree of dominance of $\overrightarrow{X_j}$ over $\overrightarrow{X_k}$ with a probabilistic estimate.

In [191], the dominance probability is designed by first evaluating the possible fitness intervals of $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$, represented by $[f_i^{\min}(\overrightarrow{X_j}), f_i^{\max}(\overrightarrow{X_j})]$ and $[f_i^{\min}(\overrightarrow{X_k}), f_i^{\max}(\overrightarrow{X_k})]$ for all objectives i=[1, N]. The probability of $\overrightarrow{X_j}$ dominating $\overrightarrow{X_k}$ is given by

$$p(\overrightarrow{X_j} \prec \overrightarrow{X_k}) = \prod_{i=1}^{N} p(f_i(\overrightarrow{X_j}) < f_i(\overrightarrow{X_k})) \tag{112}$$

where

$$p(f_i(\overrightarrow{X_j}) < f_i(\overrightarrow{X_k})) = 0, \quad \text{if } f_i^{\min}(\overrightarrow{X_j}) > f_i^{\max}(\overrightarrow{X_k}) \tag{113.a}$$

$$p(f_i(\overrightarrow{X_j}) < f_i(\overrightarrow{X_k})) = 1, \quad \text{if } f_i^{\min}(\overrightarrow{X_j}) < f_i^{\max}(\overrightarrow{X_k}) \tag{113.b}$$

Otherwise,

$$p(f_i(\overrightarrow{X_j}) < f_i(\overrightarrow{X_k})) = \frac{1}{f_i^{\max}(\overrightarrow{X_j}) - f_i^{\min}(\overrightarrow{X_j})} \int_{y=\min(f_i^{\min}(\overrightarrow{X_j}),\, f_i^{\min}(\overrightarrow{X_k}))}^{f_i^{\min}(\overrightarrow{X_k})} dy + \int_{y=\max(f_i^{\min}(\overrightarrow{X_j}),\, f_i^{\min}(\overrightarrow{X_k}))}^{\min(f_i^{\max}(\overrightarrow{X_j}),\, f_i^{\max}(\overrightarrow{X_k}))} \frac{f_i^{\max}(\overrightarrow{X_k}) - y}{f_i^{\max}(\overrightarrow{X_k})) - f_i^{\min}(\overrightarrow{X_k}))} dy \tag{113.c}$$

The following observations can be made from the above.

1. $p(\overrightarrow{X_j} \prec \overrightarrow{X_k})$=0 if $f_i^{\min}(\overrightarrow{X_j}) > f_i^{\max}(\overrightarrow{X_k})$ for any objective i∈[1, N].
2. $p(\overrightarrow{X_j} \prec \overrightarrow{X_k})$=1 if $f_i^{\max}(\overrightarrow{X_j}) < f_i^{\min}(\overrightarrow{X_k})$ for all objectives i=[1, N].
3. $p(\overrightarrow{X_j} \prec \overrightarrow{X_k}) \in (0, 1)$ if

(a) $f_i^{\min}(\overrightarrow{X_j}) < f_i^{\max}(\overrightarrow{X_k})$ for all objectives i=[1, N] and
(b) $f_i^{\max}(\overrightarrow{X_j}) < f_i^{\min}(\overrightarrow{X_k})$ for at least one objective i∈[1, N].

However, for normal distribution of $f_i(\overrightarrow{X_j}) \sim N(\overline{f}_i(\overrightarrow{X_j}), \sigma^2(f_i(\overrightarrow{X_j})))$, the dominance probability $p(f_i(\overrightarrow{X_j}) < f_i(\overrightarrow{X_k}))$ can also be modified by following the approach, proposed in [91]. Instead of considering the fitness intervals, the probabilistic estimate here is determined based on the effective fitness estimate (for example, centroid, or expected or average measure of fitness samples) and the fitness variances of the competitors $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$.

$$p(f_i(\overrightarrow{X_j}) < f_i(\overrightarrow{X_k})) = \frac{1}{2}(1 + erf(m/\sqrt{2}\, s)) \tag{114}$$

where

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{t^2} dt \tag{115}$$

$$m = \overline{f}_i(\overrightarrow{X_k}) - \overline{f}_i(\overrightarrow{X_j}) \tag{116}$$

$$s^2 = \sigma^2(f_i(\overrightarrow{X_j})) + \sigma^2(f_i(\overrightarrow{X_k})). \tag{117}$$

An alternative formulation of dominance probability is found in [156]. It is represented as

$$p(\overrightarrow{X_j} \prec \overrightarrow{X_k}) = \prod_{i=1}^{N} p(\overline{f}_i(\overrightarrow{X_j}) \le \overline{f}_i(\overrightarrow{X_k})) \times p(\overline{f}_i(\overrightarrow{X_j}) < \overline{f}_i(\overrightarrow{X_k})) \tag{118}$$

where

$$p(\overline{f}_i(\overrightarrow{X_j}) \le \overline{f}_i(\overrightarrow{X_k})) = 1 - \frac{1}{1 + \exp(-c(\overline{f}_i(\overrightarrow{X_j}) - \overline{f}_i(\overrightarrow{X_k})))} \tag{119}$$

and

$$p(\overline{f}_i(\overrightarrow{X_j}) < \overline{f}_i(\overrightarrow{X_k})) = \frac{1}{1 + \exp(-c(\overline{f}_i(\overrightarrow{X_k}) - \overline{f}_i(\overrightarrow{X_j})))}. \tag{120}$$

The proposed probabilistic dominance criterion helps us to sustain the optimal Pareto front up to certain degree of confidence. The Fermi–Dirac probability distribution above ensures that

1. For c approaching ∞ and $\overline{f}_i(\overrightarrow{X_k}) > \overline{f}_i(\overrightarrow{X_j})$ for i=[1, N], $p(\overrightarrow{X_j} \prec \overrightarrow{X_k}) = 1$ indicating $\overrightarrow{X_j} \prec \overrightarrow{X_k}$.
2. For c approaching ∞ and $\overline{f}_i(\overrightarrow{X_k}) < \overline{f}_i(\overrightarrow{X_j})$ for i=[1, N], then $p(\overrightarrow{X_j} \prec \overrightarrow{X_k}) = 0$ indicating $\overrightarrow{X_k} \prec \overrightarrow{X_j}$.
3. If $\overline{f}_i(\overrightarrow{X_j}) \cong \overline{f}_i(\overrightarrow{X_k})$ for i=[1, N], then $p(\overrightarrow{X_j} \prec \overrightarrow{X_k}) \cong 1/4^N$ indicating the non-dominance relationship between $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$.

6.7. Probabilistic truncation of extended population

In a non-dominated sorting induced multi-objective optimization, one crucial factor is to judiciously identify the qualitative diverse non-dominated solutions from the same rank population pool to maintain uniform population size. The crowding distance metric is most popularly used as a diversity preserving operator in such scenario. The solutions with higher crowding distance are preferred from their same ranked candidate pool for promotion to the next generation. The crowding distance based deterministic truncation of a Pareto front may lead to sub-optimal Pareto front in presence of stochastic noise in the fitness estimates. To alleviate this, the concept of crowding distance-

based truncation of extended population is extended by a probabilistic estimate p_j, which captures the quality of a trial solution $\overrightarrow{X_j}$. Hence, under the composite control of both p_j and the crowding distance measure $CD(\overrightarrow{X_j})$, promotion of quality and diverse solutions from the same rank candidate pool is supposed to be less misled by noise.

In [156], p_j is influenced by dominance probability. The probability that at least one population member is dominated by $\overrightarrow{X_j}$ is given by

$$p_j = 1 - \prod_{\substack{k=1,\\ k\neq j}}^{|P|} p(\overrightarrow{X_k} \prec \overrightarrow{X_j}) \tag{121}$$

with $|P|$ denotes the size of the combined population. Here, it is apparent that less the probability of $\overrightarrow{X_j}$ dominated by other population members, better is its quality. The dominance probability has already been discussed.

In [155], p_j is defined as the probability of non-occurrence of rare samples of $f_i(\overrightarrow{X_j})$ for i=[1, N]. It thus provides a measure of the degree of reliability on the samples of $f_i(\overrightarrow{X_j})$, as follows:

$$p_j = \prod_{i=1}^{N} 1 - |\gamma_{i,j}|. \tag{122}$$

where

$$\gamma_{i,j} = \frac{(Q_{i,0.25}(\overrightarrow{X_j}) - \overline{f}_i(\overrightarrow{X_j})) - (\overline{f}_i(\overrightarrow{X_j}) - Q_{i,0.75}(\overrightarrow{X_j}))}{Q_{i,0.75}(\overrightarrow{X_j}) - Q_{i,0.25}(\overrightarrow{X_j})} \tag{123}$$

is the quartile skewness that provides a robust measure of the degree of asymmetry of the distribution of fitness samples of $f_i(\overrightarrow{X_j})$ with respect to the effective fitness estimate value $\overline{f}_i(\overrightarrow{X_j})$ for i=[1, N]. The rare occurrence of fitness samples on the left (or right) tail of fitness sample distribution is indicated by $\gamma_{i,j}$ approaching −1 (or +1). These intermittent samples, far away from $\overline{f}_i(\overrightarrow{X_j})$, are supposed to be infected by noise. Contrarily, if $f_i(\overrightarrow{X_j})$ is less affected by noise, $\gamma_{i,j}\approx 0$, implying that the fitness samples are close to $\overline{f}_i(\overrightarrow{X_j})$. Thus a smaller value of $|\gamma_{i,j}|$ for i=[1, N] (or a higher value of p_j) guarantees the close proximity of the measured fitness samples (probably less noisy) to the effective fitness estimate $\overline{f}_i(\overrightarrow{X_j})$ in the sample space.

Once the quality of a trial solution $\overrightarrow{X_j}$ is captured by p_j, its diversity measure is modeled by its normalized crowding distance, denoted by

$$\overline{CD}(\overrightarrow{X_j}) = \frac{CD(\overrightarrow{X_j})}{\sum_{\forall k} CD(\overrightarrow{X_k})} \tag{124}$$

where $\overrightarrow{X_j}$ and $\overrightarrow{X_k}$ lie in the same Pareto front. Now, treating $\overline{CD}(\overrightarrow{X_j})$ like probability and presuming that p_j and $\overline{CD}(\overrightarrow{X_j})$ are independent, the selection probability of $\overrightarrow{X_j}$ for the next generation as

$$ps_j = p_j \times \overline{CD}(\overrightarrow{X_j}). \tag{125}$$

A higher value for either p_j or $\overline{CD}(\overrightarrow{X_j})$ or both increases the probability of selecting $\overrightarrow{X_j}$ from the same rank candidate pool during the truncation of extended population.

6.8. Likelihood correction

The potential of *restricted Boltzmann machine* (RBM)-based EDA in solving noisy optimization problem is studied in [172]. The marginal probability distribution of decision variables are obtained by training RBM using *constructive divergence* method. However, the so far developed probabilistic model of RBM fails to capture the real distribution of decision variables of the best solution. This problem is circumvented by applying a *likelihood correction*. The proposed correction strategy is based on the supposition that the close proximity between the RBM-predicted and the true marginal probability distributions of decision variables reduces the probability of erroneous selection of respective solutions. Moreover, the explorative capability of REDA is enhanced by hybridizing it with PSO.

7. Benchmark problems and performance metrics

The most exigent issue in validating the performance of an optimization algorithm is to identify the right benchmark functions with diverse characteristics, such as multimodality, deception, and isolation, to resemble complicated real world problems. The most well known benchmark problems for noisy optimization are recommended in GECCO'2010, referred to as BBOB benchmark suite [73]. Among the test suite of 30 benchmark functions, six functions (f101–f106) are affected with moderate noise, fifteen (f107–f121) with severe noise and the rest (f122–f130) are highly multi-modal functions with severe noise. Three different noise models are used in the benchmark suite, including the Gaussian, the uniform, and the Cauchy noise models. Among these three models, the first two represent the multiplicative models, while the third one is devised as an additive noise model. The noise models are developed assuming positive objective function values in all circumstances. The definitions of the noise models are reproduced below for the sake of completeness of the paper.

(i) Gaussian noise model: It is defined as

$$f_{noisy.G}(\overrightarrow{X}) = f(\overrightarrow{X}) \times \exp(\beta \times \mathrm{N}(0, 1)) \tag{126}$$

for a trial solution $\overrightarrow{X}$. Here the contamination level of noise is controlled by $\beta\in$[0.01, 1].

(ii) **Uniform noise model**: It is defined as

$$f_{noisy.U}(\overrightarrow{X}) = f(\overrightarrow{X}) \times \mathrm{U}(0, 1)^{\beta} \times \max\left[1, \left(\frac{10^9}{f(\overrightarrow{X}) + \varepsilon}\right)^{\alpha\times \mathrm{U}(0,1)}\right] \tag{127}$$

where U(0, 1) represents a random number, uniformly distributed in (0, 1). The constant parameters α and β control the jeopardizing strength of noise. In a D-dimensional search space, α is set as $\beta\times(0.49+1/D)$ where β=0.01 and β=1 for moderate and severe noise respectively. ε is a positive constant, as small as 10^{-99}.

(iii) **Cauchy noise model:** It is defined as

$$f_{noisy.C}(\overrightarrow{X}) = f(\overrightarrow{X}) + \alpha \times \max\left[0, 1000 + \mathbf{I}_{\{\mathrm{U}(0,1)<p\}}\frac{\mathrm{N}(0, 1)}{|\mathrm{N}(0, 1)| + \varepsilon}\right] \tag{128}$$

where $\alpha\in$[0.01, 1] governs the noise strength and $p\in$[0.05, 0.2] represents the frequency of noise disturbance. The most challenging issue to handle such noise is to identify and suitably treat the large *outliers*, originating from a continuous distribution.

To compare the relative merits of different noisy EAs to minimize the benchmark functions of the BBOB test suite, *expected run-time* (*ERT*) is considered as the most appropriate performance metric. *ERT* of an EA is defined as the expected number of objective function evaluations to achieve a target value of a benchmark function for the first time. It is mathematically given as

$$ERT = RT_S + \frac{1 - p_s}{p_s} RT_{US} \tag{129}$$

where the run-times RT_S and RT_{US} respectively symbolize the average number of objective function evaluations for successful and unsuccessful runs of an algorithm. First, an EA is run on each benchmark function of BBOB framework and is terminated either on achieving the target objective function value or reaching the maximum number of

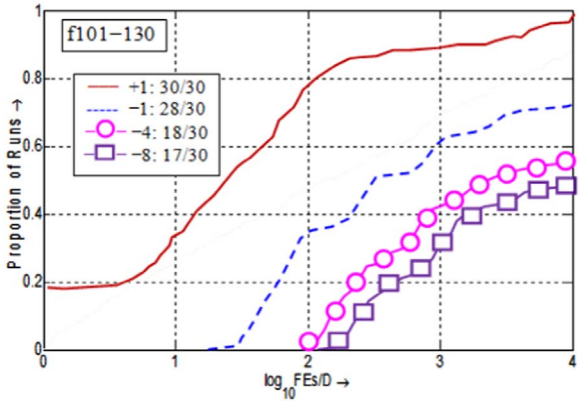

Fig. 12. *ECDF* of the *ERT* (number of function evaluations, divided by search space dimension *D*) of a noisy evolutionary optimization algorithm with target objective function value= $f^{opt}+\Delta f$ with f^{opt} as the optimum objective function value and $\Delta f = 10^k$, where k=1; −1; −4; −8 (the first value in the legend) while the second value in the legend indicating the number of functions (out of 30) that were solved in at least one run.

objective function evaluations ($10^4 \times D$ for *D*-dimensional problem), whichever occurs earlier. Out of such *N* runs of a noisy EA on each BBOB benchmark problem, the runs, where the EA remains successful in achieving the target value, are referred to as successful runs. The non-zero success rate p_s represents the fraction of successful runs of the corresponding EA.

The *ERT* of an EA is obtained from the run-times of its *N* optimization runs on a BBOB benchmark problem. The run-times corresponding to these *N* optimization runs form a "single data sample set". Bootstrapping is additionally used as a tool in BBOB framework to effectively capture the true *empirical cumulative distribution function* (*ECDF*) of *ERT*. To obtain the bootstrapped distribution of *ERT*, a large number (for example, 100) of such "single data sample set"s is resampled. Each "single data sample set" here is constructed by repeatedly drawing single optimization runs (of the EA on each benchmark function) with replacement until a successful run is drawn. The run-time in each "single data sample set" (in the bootstrapped distribution of *ERT*) is computed as the sum of the number of function evaluations of its constituent optimization runs. The *ECDF* of the *ERT* of a noisy EA is pictorially represented in Fig. 12.

In [154], the authors have validated the efficacy of their proposed noisy multi-objective optimization algorithm on the noisy versions of the CEC'2009 benchmark suite [201]. Each of the 23 benchmark functions of the test suite is contaminated with additive noise samples taken from the Gaussian, the Poisson and the random distribution, as tabulated in Table 4.

The following performance metrics are used for comparative analysis of the performances of the state-of-art noisy multi-objective optimization (MOO) algorithms.

(a) **Inverted generational distance (*IGD*):** The *IGD* between an approximate Pareto front *A* (obtained by an MOO algorithm) and the true optimal Pareto front *P* of an MOO benchmark problem (in objective space) is defined as

$$IGD(A, P^*) = \frac{\sum_{z \in P^*} d(z, A)}{|P^*|} \tag{130}$$

with $d(v, A)$ denoting the minimum Euclidean distance between z and the points in A, both in the objective space. Lower the *IGD* measure, closer is the approximate Pareto front A to the optimal Pareto front P^*.

(b) **Spacing (Δ):** The metric provides a unique measure of the spread and distribution of the trial solutions in the objective surfaces of the MOO problem. The metric is defined as follows:

$$\Delta = \sqrt{\frac{1}{M-1}\sum_{j=1}^{M}(\bar{d} - d_j)^2}, \quad \bar{d} = \frac{1}{M}\sum_{j=1}^{M} d_j$$
$$\text{and } d_j = \min_{\substack{k=1,\\ k \neq j}}^{M} \sum_{i=1}^{N} |f_i(\vec{X_j}) - f_i(\vec{X_k})| \quad \text{for } i = [1, M]. \tag{131}$$

Here M represents the number of non-dominated trial solutions found by the MOO algorithm with $\vec{X_j}$ and $\vec{X_k}$ as two non-dominated members of the approximate Pareto front A. An ideal value of zero for **Δ** signifies all the members of A are equidistantly spaced.

(c) **Error ratio (*ER*):** This metric is defined as

$$ER = \frac{\sum_{j=1}^{M} e_j}{M}, \quad e_j = \begin{cases} 0, & \text{if } \vec{X_j} \in A \text{ and } \vec{X_j} \in P^*, \\ 1, & \text{if } \vec{X_j} \in A \text{ and } \vec{X_j} \notin P^*. \end{cases} \tag{132}$$

An ideal value of zero for *ER* indicates that all the non-dominated members of A also belong to the optimal Pareto front P^*.

(d) **Hypervolume ratio (*HVR*):** The size of the objective spaces covered by a set S of non-dominated solutions (of an MOO problem) is referred to as its hypervolume $HV(S)$. Based on this definition, *HVR* is defined as follows.

$$HVR(A) = \frac{HV(A)}{HV(P^*)}. \tag{133}$$

Intuitively, $HVR(A)$ achieves its ideal value 1 if the non-dominated members of A (in the objective spaces) are same as the members of P^*. A focused HVR (*F-HVR*) metric is also proposed in [177] to assess the convergence speed and diversity of the final population obtained by a noisy R-NSGA-II algorithm variant [61].

(e) **Population diversity:** In [176], population diversity is used as a performance metric to validate the performance noisy MOO algorithms. It is defined as follows.

$$PD(P) = \sum_{\vec{X} \in P} CD(\vec{X})/|P|. \tag{134}$$

Here $|P|$ denotes the size of the final population P and $CD(\vec{X})$ represents the crowding distance measure of the trial solution $\vec{X}$. Intuitively, higher the measure of $PD(P)$, better is the performance of the noisy MOO algorithm to preserve the population diversity.

In [178], three different noise models are considered to contaminate the fitness of population members with the noise samples. The

Table 4
Probability distribution functions of injected noise η.

Noise model	PDF	Distribution properties	Method
Gaussian	$\exp(-(\eta-\mu)^2/2\sigma^2)/\sigma\sqrt{2\pi}$	mean= μ, variance= σ^2	Box–Muller [34]
Poisson	$\lambda^{\eta}\exp(-\lambda)/\eta!$	mean= λ, variance=λ	Knuth [106]
Random	Random		Linear congruential pseudo random number generator [31]

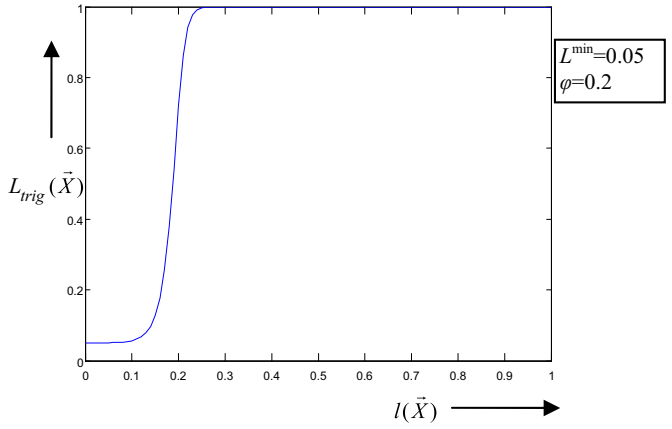

Fig. 13. Logistic-growth noise model.

amplitude of the noise samples are determined based on the distance $g(\overrightarrow{X})$ [203] of the trial solution $\overrightarrow{X}$ from the optimal Pareto front. The normalized value of $g(\overrightarrow{X})$ is given by

$$l(\overrightarrow{X}) = \frac{g(\overrightarrow{X}) - 1}{g^{\max}(\overrightarrow{X}) - 1},\ g^{\max}(\overrightarrow{X}) = \max_{\forall \overrightarrow{X} \in P} g(\overrightarrow{X}) \tag{135}$$

The three noise models adopted in [178] based on $l(\overrightarrow{X})$ are given as follows.

(i) **Logistic-growth noise model**: It is defined as

$$f_{noisy.\log}(\overrightarrow{X}) = f(\overrightarrow{X}) + \mathrm{N}(0, L_{\log}(\overrightarrow{X}) \times \sigma) \tag{136}$$

where,

$$L_{\log}(\overrightarrow{X}) = \frac{1 - L^{\min}}{(1 + \exp(-100(l(\overrightarrow{X}) - \phi)))^{0.5}} + L^{\min} \tag{137}$$

for a trial solution $\overrightarrow{X}$. Here $L^{\min}$ is the minimum noise level close to the optimal Pareto front and σ represents the predefined standard deviation of the objective function. As evident from (137) and Fig. 13, the noise level quickly attains its maximum value of unity following a logistic-growth function around the threshold value φ. The jeopardizing effect of noise taken from the logistic-growth model imposes challenges for the preference-based algorithms that determine the sampling requirements based on the distance of the population members from the reference point.

(i) **Trigonometric noise model:** It resembles a co-sinusoidal noise landscape as shown in Fig. 14. It is defined as

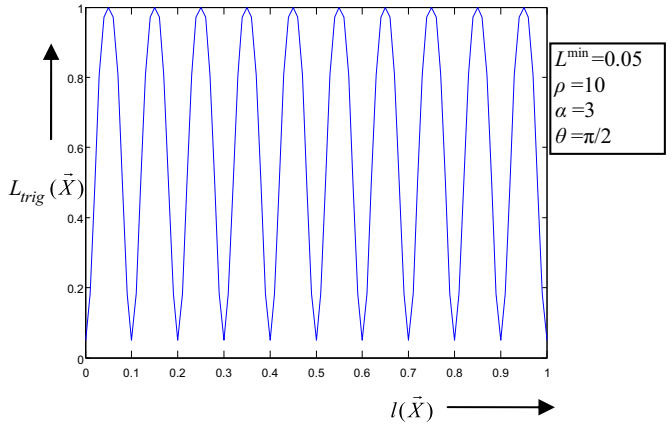

Fig. 14. Trigonometric noise model.

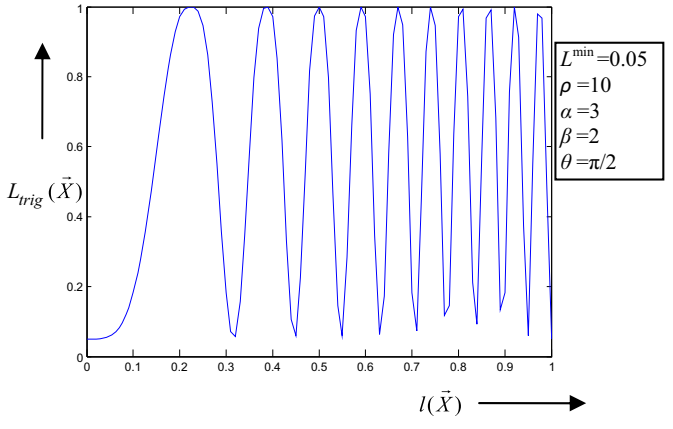

Fig. 15. Delayed trigonometric noise model.

$$f_{noisy.trig}(\overrightarrow{X}) = f(\overrightarrow{X}) + \mathrm{N}(0, L_{trig}(\overrightarrow{X}) \times \sigma) \tag{138}$$

where,

$$L_{trig}(\overrightarrow{X}) = -(1 - L^{\min})|\sin(\rho\pi l(\overrightarrow{X}) - \theta)|^{\alpha} + 1 \tag{139}$$

where ρ denotes the number of peaks and α is a constant parameter controlling the width of the peaks. The levels of noise samples contaminating the objective function values of the trial solutions residing at different Pareto fronts can be controlled by varying the phase parameter θ. The most challenging issue to handle such noise lies in dealing with its large number of hills and valleys, as shown in Fig. 14.

(i) **Delayed trigonometric noise model:** It is defined as

$$f_{noisy.delay.trig}(\overrightarrow{X}) = f(\overrightarrow{X}) + \mathrm{N}(0, L_{delay.trig}(\overrightarrow{X}) \times \sigma) \tag{140}$$

where,

$$L_{delay.trig}(\overrightarrow{X}) = -(1 - L^{\min})|\sin(\rho\pi l(\overrightarrow{X})^{\beta} - \theta)|^{\alpha} + 1 \tag{141}$$

where ρ denotes the number of peaks and α is a constant parameter controlling the width of the peaks. The designed noise model ensures low level of noise close to the optimal Pareto front. The size of the low-noise area close to the optimal Pareto front is determined by the setting of the parameter β. Thus, the noise model imposes the challenges of both the logistic-growth and the trigonometric models, discussed earlier (Fig. 15).

8. Open problems and discussion

This section includes a brief discussion on the NOPs. A few promising future research directions are also given in this section.

8.1. Discussion

The infiltration of noise in an evolutionary optimization algorithm can be avoided at five different levels: (i) sampling, (ii) fitness estimation, (iii) population sizing, (iv) evolutionary search operation and (v) selection. At the sampling level, the sample size determination is an important issue as it greatly influences the effect of inducing noise at the subsequent steps of the algorithm. At the fitness estimation level, the effective fitness of a trial solution is estimated from the noisy samples of the same solution by a suitable aggregation operation on the sample fitness values. The third strategy deals with increasing the population size to reduce the impact of noise on selection operation. Apparently, the optimal selection of the sample size per trial solution and the entire population size is constrained by the computational complexity of solving a given real world noisy optimization process.

Table 5
Favorable conditions for different noise handling strategies.

Strategy	Condition		Action	
Sample size bound	**Population size**	**Search dimension**	n^{min}	$ss=n^{max}-n^{min}$
	Large	Low	Small	Small
	Large	High	Moderate	Small
	Small	High	Large	Large
	Small	Low	Small	Moderate
Fitness estimation	**Noise variance**	**Fitness landscape**	**Fitness estimation policy**	
	Small	Unimodal	Averaging	
	Small	Multimodal	Expectation (uniform partitioning of sample space)	
	Large	Unimodal	Expectation (non-uniform partitioning of sample space)	
	Large	Multimodal	Centroidal defuzzification	
Parameters of evolutionary search dynamics	**Fitness landscape**		**Control parameter settings**	
	Unimodal		Fixed parameter values	
	Multimodal		Online tuning using local search	
Selection Criteria	**Fitness landscape**		**Selection policy**	
	Unimodal		Statistical test based selection	
	Multimodal		Probabilistic selection	

Improved evolutionary search operators (for example, mutation, and crossover) are concerned with directing the candidate solutions towards the truly potential zones in the noise-induced fitness landscape. Lastly, the selection helps filtering of quality solutions from the rest. A theoretical insight to the existing literature on noisy evolutionary optimizations reveals that their performances greatly rely on the characteristics of the fitness landscape and injected noise. This section unfolds the modalities of the previously mentioned five strategies in eliminating the effect of noise infiltration in different objective surfaces as given in Table 5.

As evident from the existing sampling strategies, a possible trade-off between the computational complexity and the accurate inspection of noise contamination can be efficiently balanced by dynamic sampling. The performance of the dynamic strategy, however, is dependent on the appropriate settings of minimum sample size (n^{min}) and the possible sample size span ($ss=n^{max}-n^{min}$). Apparently, for a large population size, the detrimental effect of noise on a population member can be compensated by the average behavior of similar members due to inherent search capability of EAs. In such scenario, it is obvious to use a small value for both n^{min} and ss. This is particularly effective in a low-dimensional search space, densely populated by candidate solutions. If the search space dimension is high, a large population of trial solutions with non-uniform distribution over the wide parameter space may not be able to examine precisely the penetration of noise in their fitness measures. This demands a moderate value of n^{min}. Apparently, a large setting of n^{max} is anticipated to increase the computational burden without significantly enhancing the quality of noise elimination. Hence, ss (>0) is kept small. However, for a high-dimensional problem with low population size, the population members are sparsely distributed over the parameter space. Hence, a specific trial solution is used to represent a relatively large area in the search space. A large value for both n^{min} and ss, under this circumstances is expected to extract the true fitness information in the realm (within the search space) of the given trial solution. Contrarily, a small population in a low dimensional search space can proficiently capture the local distribution of noise using small sample size per solution.

Typical aggregation operators used to derive the effective fitness estimate of a trial solution from its fitness samples include averaging, expectation and centroidal de-fuzzification. The efficiency of fitness estimators is contingent upon its capability to capture the local noise distribution among the fitness samples. In presence of uniform local noise distribution of low variance, averaging is usually preferred to reduce the computational complexity. However, the large scale jeopardizing effect of non-uniform noise distribution can be competently coped with expectation and centroidal de-fuzzification. The centroidal de-fuzzification outperforms the expectation policy (uniform or non-uniform partitioning of the sample space) in modelling the noise-induced uncertainty in the occurrence of fitness samples, however, at the cost of additional computational complexity. It is worth mentioning that the superiority of the de-fuzzification strategy over the expectation is significantly manifested for multimodal fitness landscapes.

The adverse consequence of noise becomes prominent when injected into multimodal objective surface. Under such circumstances, mere applications of the dynamic sampling (with moderately high sampling rate) and the centroidal de-fuzzification stratagems fail to precisely locate the true global optimum. A possible solution to the failure of discovering promising area in the fitness landscape is to autonomously adapt the control parameters of evolutionary search dynamics by learning the noise-induced objective surface characteristics. The online tuning strategy is particularly essential when the amplitude and frequency of noise occurrence are high. The constant settings of control parameters may remain successful in avoiding the deceptive effect of noise infiltration in unimodal fitness landscape when used along with proficient strategies of sampling and fitness estimation.

Lastly, the robustness of selection process against noise is one important issue in ensuring the survival of quality solutions over generations. Two categories of noise-tolerant selection strategies have gained wide publicity due to their performances: statistical test based selection and probabilistic selection. The performance of the statistical test based selection, however, greatly relies on the setting of the confidence interval (or the restriction parameters in [154]). Moreover, the competitive sampling policy, when realized with the statistical test, augments the computational overhead. Hence, this strategy is presumed to perform efficiently and time-optimally in unimodal fitness landscape. The probabilistic selection strategy reduces the run-time complexity of the algorithm at the cost of reduction in the certainty level of sustaining quality solutions.

8.2. Open problems

Although there exists quite a few interesting papers on noisy evolutionary optimization, there remains immense scope for future researchers in this new domain of research. The following open problems are of great importance to enhance the performance of the noisy optimization problems.

1) *Learning based adaptive sample size selection*: Existing noisy evolutionary optimization algorithms primarily consider noise variance as a metric to determine the sample size of a trial solution. For large polulation size, the existing methods of sample size selection is acceptable. However, when the populaton size is small (with respect to the search space dimension), it becomes difficult to get an approximate idea of the noisy surface from the few population members spread across the surface.

   An alternative apoproach could be to learn the objective surface approximately by evaluating the objective function of the population in the local neighborhood. In case the fitness measure in the local neighborhood has a wide variance, the sample size of the members falling in the local neighborhood would be high. Any traditional learning algorithms could apparently be used to handle the above situation. For instance, a supervised learning algorithm can be trained with local fitness variation as input and sample size as output. After the learning is over, the learning system could be used to predict sample size from local fitness variance. However, the training-time of supervised learning algorithm being high, the run-

time complexity of the noisy optimization algorithm may increase significantly. The situation would be similar for unsupervised clustering based sample size selection and thus is not suggestive. The other problem that might appear due to learning based sample size selection is repeated learning requirement as the surface might be corrupted with time-varying noise. The only solution that could eliminate the last issue is the incorporation of reinforcement learning (RL) in sample size selection.

In RL, we usually store the reward/penalty earned by an agent at distinct system states for a distinct action [138,200]. In the present context, the quantized intervals of the range of fitness variance are used as states and the sample sizes as actions. The reward/penalty stored in the grids, indexed by state-action pairs, are representative of the performance of the action at the selected state with respect to a fixed goal, here finding optimum fitness. The advantage of this learning paradigm lies in its potential for life-long learning in real-time. Naturally, the best action at a given state can be selected using the recent learning instances about the objective surface. The open problem here rests on the choice of reward function, state-definition based on the quantization procedure and the learning strategy based on the measure of reward/penalty.

2) *Uncertainty modelling in effective fitness evaluation*: The fitness estimate of the trial solutions in noisy optimization problem suffers from measurement noise, and thus the true estimate of the fitness from noisy samples requires additional processing. The motivation of the additional processing is to eliminate the effect of noise from noisy fitness values. This, in turn, is an uncertainty manangement problem, which could be addressed by several well-known techniques, including stochastic, fuzzy/rough set based models and Dempster-Shafer formulation. There exist much scope for future research on accurate fitness estimation problem in presence of noise.
3) *Dynamic population sizing*: An alternative approach to handle noisy optimization problem is to vary the population size instead of sample size as adopted in the traditional literatuire. There remains open ends on the functional form of population-size with (global) noise variance on the landscape.
4) *Adaptation of evolutionary search strategy*: To overcome the deception by noise, existing noisy optimization techniques consider effective exploration of the search space in the earlier phase of the algorithm, while fine tuning the search in the local neighborhood of quality solyutions in the exploitation phase. There exists extensive research openings to autonomously learn the control parameters with an aim to select right set of parameters for a set of local population members on the serach landscape.
5) *Improving selection strategy*: In traditional noisy optimization algorithms stochastic selection is usuallay adopted instead of deterministic selection. However, there is a chance to lose the local optima. Local optima being highly probable regions for possible global optimum, losing local optima may render a miss of the global optimum. The problem is more severe in noisy optimization. There remains broad research opening to preserve the local optima in noisy optimization problem by introducing niching strategy [113].

9. Conclusion

Evolutionary algorithms have proven significant impact on many real world optimization problems. Noise in engineering/ scientific problems acts as a fundamental impediment to furthering research. Evolutionary algorithms that produce promisingly progressive outcome in research and practice cannot work in presence of noise on the objective function. This paper provides a thorough review to handle optimization in presence of noise on the fitness landscape. Five different strategies have been proposed to search optimal solutions on for NOPs. The strategies include sampling, fitness estimation, dynamic population sizing, adaptation of evolutionary search strategy and modification of selection strategy. Details of these are given in the respective sections and are not reproduced here for page restriction.

Several researchers have put forward different noise models for their specific noisy optimization problems. Although the solutions proposed by them work brilliantly for the suggested noise models, they occasionally do not apply well for different noise models. Consequently, there is hardly any universally acceptable algorithm to handle noisy optimization problems in general. This motivated researchers to take up hybrid models that can cope up to different noise models.

There still remain quite a few unattended areas in noisy evolutionary optimization algorithms. A few of these that need special attention are included as open problems for future research. One genuinely attractive issue, which has not received enough popularity, to the best of these authors' knowledge, is introducing learning in the optimization algorithm, primarily to learn the objective surface for different parametric space. Although any traditional learning strategy could have been incorporated, the authors here justified the importance of reinforcement learning over its two main counterparts, supervised and unsupervised learning, because of the real-time learning characteristics of the former learning algorithms. Two distinct subproblems, where reinforcement learning could be incorporated include sampling and control parameter adaptation of the evolutionary algorithms.

Noisy optimization algorithms would serve a remedy for many engineering optimization problems in robotics, material processing, control of missiles, autonomous vehicle parking and many others. In robotics, noisy optimization influences sensory measurements, which in turn influence the decisions in intelligent planning concerning optimal path, optimal time to reach target and/or team cooperation among the robot team-mates. In material processing, noise in sensors induces noise in the decisions and thus influences the material processing, which in turn hampers the quality of the finished product. In control application of surface to surface missiles, atmospheric storm may appear at any point of time at any arbitrary direction. Controller tuning in presence of random noise to eliminate possible path deviation of the missile thus is an important real-time optimization problem. Autonomous vehicle parking in presence of sensory noise may cause collision of the vehicles, where noisy evolutionary real-time optimization plays an important role.

Acknowledgment

Funding by University Grants Commission in the Project: University with Potential for Excellence Program in Cognitive Science is gratefully acknowledged for the present work.

References

[1] B.M. Adams, Advanced topics in statistical process control: the power of Shewhart's charts, Technometrics 38 (3) (1996) (286–286).
[2] A. N.Aizawa, B. W.Wah, Dynamic control of genetic algorithms in a noisy environment, in: Proceedings of the Fifth International Conference on Genetic Algorithms, vol. 2, 1993, p. 1
[3] A.N. Aizawa, B.W. Wah, Scheduling of genetic algorithms in a noisy environment, Evolut. Comput. 2 (2) (1994) 97–122.
[4] S.B. Akat, V. Gazi, Particle swarm optimization with dynamic neighborhood topology: three neighborhood strategies and preliminary results, Proc. IEEE Swarm Intell. Symp. (2008) 1–8.
[5] Y. Akimoto, S.A. Morales, O. Teytaud, Analysis of runtime of optimization algorithms for noisy functions over discrete codomains, Theor. Comput. Sci. 605 (2015) 42–50.
[6] D.V. Arnold, H.G. Beyer, A comparison of evolution strategies with other direct search methods in the presence of noise, Comput. Optim. Appl. 24 (1) (2003) 135–159.
[7] D.V. Arnold, H.G. Beyer, A general noise model and its effects on evolution strategy performance, IEEE Trans. Evolut. Comput. 10 (4) (2006) 380–391.
[8] D.V. Arnold, H.G. Beyer, Efficiency and mutation strength adaptation of the (μ/μ I, λ)-ES in a noisy environment, *Parallel Probl. Solving Nat.*, Springe. Berl. Heidelb. (2000) 39–48.
[9] D.V. Arnold, H.G. Beyer, Local performance of the (1+1)-ES in a noisy environ-

ment, IEEE Trans. Evolut. Comput. 6 (1) (2002) 30–41.
[10] D.V. Arnold, H.G. Beyer, On the benefits of populations for noisy optimization, Evolut. Comput. 11 (2) (2003) 111–127.
[11] D.V. Arnold, H.G. Beyer, Performance analysis of evolution strategies with multi-recombination in high-dimensional RN-search spaces disturbed by noise, Theor. Comput. Sci. 289 (1) (2002) 629–647.
[12] D.V. Arnold, H.G. Beyer, Performance analysis of evolutionary optimization with cumulative step length adaptation, IEEE Trans. Autom. Control 49 (4) (2004) 617–622.
[13] D.V. Arnold, H.G. Beyer, Investigation of the (μ, λ)-ES in the presence of noise, Proc. IEEE Congr. Evolut. Comput. 1 (2001) 332–339.
[14] M. Babbar, A. Lakshmikantha, D.E. Goldberg, A modified NSGA-II to solve noisy multiobjective problems, in: Proceedings of Genetic and Evolutionary Computation Conference. Late-Breaking Papers, 2003, pp. 21–27.
[15] T. Bäck, U. Hammel, Evolution strategies applied to perturbed objective functions, Proc. IEEE Congr. Evolut. Comput. (1994) 40–45.
[16] T. Bäck, Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms, Oxford University Press, 1996.
[17] P.G. Balaji, D. Srinivasan, C.K. Tham, Uncertainties reducing Techniques in evolutionary computation, Proc. IEEE Congr. Evolut. Comput. (2007) 556–563.
[18] M. Basseur, E. Zitzler, A preliminary study on handling uncertainty in indicator-based multiobjective optimization, EvoWorkshops (2006) 727–739.
[19] M. Basseur, E. Zitzler, Handling uncertainty in indicator-based multiobjective optimization, Int. J. Comput. Intell. Res. 2 (3) (2006) 255–272.
[20] H.G. Beyer, Actuator noise in recombinant evolution strategies on general quadratic fitness model, in: Proceedings of Genetic and Evolutionary Computation, Springer Berlin Heidelberg, 2004, pp. 654–665
[21] H.G. Beyer, An alternative explanation for the manner in which genetic algorithms operate,, Bio Syst. 41 (1) (1997) 1–15.
[22] H.G. Beyer, Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice,, Comput. Methods Appl. Mech. Eng. 186 (2) (2000) 239–267.
[23] H.G. Beyer, Mutate large, but inherit small! On the analysis of rescaled mutations in (1, λ)-ES with noisy fitness data,, *Parallel Probl. Solving Nat.*, Springe. Berl. Heidelb. (1998) 109–118.
[24] H.G. Beyer, B. Sendhoff, Evolution strategies for robust optimization,, Proc. IEEE Congr. Evolut. Comput. (2006) 1346–1353.
[25] H.G. Beyer, B. Sendhoff, Functions with noise-induced multimodality: a test for evolutionary robust optimization-properties and performance analysis,, IEEE Trans. Evolut. Comput. 10 (5) (2006) 507–526.
[26] H.G. Beyer, D.V. Arnold, The steady state behavior of (μ/μ I, λ)-ES on ellipsoidal fitness models disturbed by noise,, *Proc. Genet. Evolut. Comput.*, Springe. Berl. Heidelb. (2003) 525–536.
[27] H.G. Beyer, H.P. Schwefel, Evolution strategies – a comprehensive introduction,, Nat. Comput. 1 (1) (2002) 3–52.
[28] H.G. Beyer, D.V. Arnold, S.M. Nieberg, A new approach for predicting the final outcome of evolution strategy optimization under noise, Genet. Program. Evol. Mach. 6 (1) (2005) 7–24.
[29] H.G. Beyer, M. Olhofer, B. Sendhoff, On the impact of systematic noise on the evolutionary optimization performance—a sphere model analysis, Genet. Program. Evol. Mach. 5 (4) (2004) 327–360.
[30] T. Blackwell, Particle swarm optimization in dynamic environmentsProceedings of Evolutionary Computation in Dynamic and Uncertain environments, Springer Berlin Heidelberg, 2007, pp. 29–49.
[31] J. Bolte, Linear congruential generators, Wolfram Demonstrations Project
[32] P. Boonma, J. Suzuki, A confidence-based dominance operator in evolutionary algorithms for noisy multiobjective optimization problems, in: Proceedings of the 21st IEEE International Conference on Tools with Artificial Intelligence, 2009, pp. 387–394.
[33] P.A.N. Bosman, J. Grahl, D. Thierens, Benchmarking parameter-free AMaLGaM on functions with and without noise, Evolut. Comput. 21 (3) (2013) 445–469.
[34] G.E.P. Box, M.E. Muller, A note on the generation of random normal deviates, Ann. Math. Stat. 29 (1958) 610–611.
[35] J. Branke, Creating robust solutions by means of evolutionary algorithms, *Proc. Parallel Probl. Solving Nat.*, Springe. Berl. Heidelb. (1998) 119–128.
[36] J. Branke, C. Schmidt, Selection in the presence of noise, Proc. Genet. Evolut. Comput., Springe. Berl. Heidelb. (2003) 766–777.
[37] J. Branke, C. Schmidt, Sequential sampling in noisy environments, Proc. Parallel Probl. Solving Nat., Springe. Berl. Heidelb. (2004) 202–211.
[38] J. Branke, C. Schmidt, H. Schmec, Efficient fitness estimation in noisy environments, Proc. Genet. Evolut. Comput. (2001) 243–250.
[39] J. Branke, S. Meisel, C. Schmidt, Simulated annealing in the presence of noise, J. Heuristics 14 (6) (2008) 627–654.
[40] J. Brest, S. Greiner, B. Bošković, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evolut. Comput. 10 (6) (2006) 646–657.
[41] J. Brownlee, Clever algorithms: nature-inspired programming recipes, Jason Brownlee, 2011.
[42] D. Buche, P. Stoll, R. Dornberger, P. Koumoutsakos, Multiobjective evolutionary algorithm for the optimization of noisy combustion processes, IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev. 32 (4) (2002) 460–473.
[43] L. T. Bui, D. Essam, H. A. Abbass, D. Green, Performance analysis of evolutionary multi-objective optimization methods in noisy environments, in: Proceedings of Asia Pacific Symposium on Intelligent And Evolutionary Systems, 2004, pp. 29–39
[44] L.T. Bui, H.A. Abbass, D. Essam, Fitness inheritance for noisy evolutionary multi-objective optimization, in: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, ACM, 2005, pp. 779–785.
[45] L.T. Bui, H.A. Abbass, D. Essam, Localization for solving noisy multi-objective optimization problems, Evolut. Comput. 17 (3) (2009) 379–409.
[46] L.T. Bui, H.A. Abbass, M. Barlow, A. Bender, Robustness against the decision-maker's attitude to risk in problems with conflicting objectives, IEEE Trans. Evolut. Comput. 16 (1) (2012) 1–19.
[47] A. Caponio, F. Neri, Differential evolution with noise analyzerProceedings of Applications of Evolutionary Computing, Springer Berlin Heidelberg, 2009, pp. 715–724.
[48] M.L. Cauwet, J. Liu, B. Roziere, O. Teytaud, Algorithm portfolios for noisy optimization, Ann. Math. Artif. Intell. 76 (1–2) (2016) 143–172.
[49] C.H. Chen, D. He, M. Fu, Efficient simulation budget allocation for selecting an optimal subset, INFORMS J. Comput. 20 (4) (2008) 579–595.
[50] C.H. Chen, J. Lin, E. Yücesan, S.E. Chick, Simulation budget allocation for further enhancing the efficiency of ordinal optimization, Discret. Event Dyn. Syst. 10 (3) (2000) 251–270.
[51] J.Y. Chia, C.K. Goh, V.A. Shim, K.C. Tan, A data mining approach to evolutionary optimisation of noisy multi-objective problems, Int. J. Syst. Sci. 43 (7) (2012) 1217–1247.
[52] S.Y. Chiu, C.N. Lin, J. Liu, T.C. Su, F. Teytaud, O. Teytaud, S.J. Yen, Differential evolution for strongly noisy optimization: Use 1.01^n resamplings at iteration n and reach the −1/2 Slope, in: Proceedings of IEEE Congress on Evolutionary Computation, 2015, pp. 338–345.
[53] M. Clerc, James Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evolut. Comput. 6 (1) (2002) 58–73.
[54] H.G. Cobb, An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments, Memorandum 6760, Naval Research Laboratory, Washington DC, 1990.
[55] D. Costa, E.A. Silver, Tabu search when noise is present: an illustration in the context of cause and effect analysis, J. Heuristics 4 (1) (1998) 5–23.
[56] P.J. Darwen, Computationally intensive and noisy tasks: co-evolutionary learning and temporal difference learning on backgammon, Proc. IEEE Congr. Evolut. Comput. 2 (2000) 872–879.
[57] P.J. Darwen, J.B. Pollack, Co-evolutionary learning on noisy tasks, in: Proceedings of IEEE Congress on Evolutionary Computation, vol. 3, 1999, pp. 1731
[58] S. Das, A. Konar, U.K. Chakraborty, Improved differential evolution algorithms for handling noisy optimization problems, Proc. IEEE Congr. Evolut. Comput. 2 (2005) 1691–1698.
[59] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art,, IEEE Trans. Evolut. Comput. 15 (1) (2011) 4–31.
[60] K. Deb, H. Gupta, Introducing robustness in multi-objective optimization,, Evolut. Comput. 14 (4) (2006) 463–494.
[61] K. Deb, J. Sundar, Reference point based multi-objective optimization using evolutionary algorithms, in: Proceedings of Genetic and EvolutionaryComputation, ACM, 2006, pp. 635–642
[62] K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, A fast and elitist multi-objective genetic algorithm: nsga-ii, IEEE Trans. Evolut. Comput. 6 (2) (2002) 182–197.
[63] J.E. Diaz, J. Handl, Implicit and explicit averaging strategies for simulation-based optimization of a real-world production planning problem, Informatica 39 (2) (2015) 161–168.
[64] G. Dueck, Gunter, Tobias Scheuer, Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing, J. Comput. Phys. 90 (1) (1990) 161–175.
[65] H. Eskandari, C.D. Geiger, Evolutionary multiobjective optimization in noisy problem environments, J. Heuristics 15 (6) (2009) 559–595.
[66] H. Eskandari, C.D. Geiger, R. Bird, Handling uncertainty in evolutionary multi-objective optimization: SPGA, Proc. IEEE Congr. Evolut. Comput. (2007) 4130–4137.
[67] M.J. Feigenbaum, Quantitative universality for a class of nonlinear transformations, J. Stat. Phys. 19 (1) (1978) 25–52.
[68] J.E. Fieldsend, Elite accumulative sampling strategies for noisy multi-objective optimisation, in: Proceedings of Evolutionary Multi-Criterion Optimization, Springer International Publishing,Publishing, 2015, pp. 172–186.
[69] J.E.Fieldsend, R.M.Everson, Efficiently identifying Pareto solutions when objective values change, in: Proceedings of Genetic and Evolutionary Computation, ACM, 2014, pp. 605–612.
[70] J.E. Fieldsend, R.M. Everson, Multi-objective optimisation in the presence of uncertainty, Proc. IEEE Congr. Evolut. Comput. 1 (2005) 243–250.
[71] J.E. Fieldsend, R.M. Everson, The rolling tide evolutionary algorithm: a multi-objective optimizer for noisy optimization problems, IEEE Trans. Evolut. Comput. 19 (1) (2015) 103–117.
[72] J.E. Fieldsend, R. M.Everson, On the efficient maintenance and updating of Pareto solutions when assigned objectives values may change, Technical Report, University of Exeter, UK, 2013.
[73] S. Finck, N. Hansen, R. Ros, A.Auger, Real-Parameter Black-Box Optimization Benchmarking 2010: Presentation of the Noisy Functions, Working Paper 2009/21, compiled December 4, 2014
[74] J.M. Fitzpatrick, J.J. Grefenstette, Genetic algorithms in noisy environments, Mach. Learn. 3 (2–3) (1988) 101–120.
[75] S.B. Gelfand, S.K. Mitter, Simulated annealing with noisy or imprecise energy measurements, J. Optim. Theory Appl. 62 (1) (1989) 49–62.
[76] B.Goethals, Survey on frequent pattern mining, University of Helsinki, 2003.
[77] C.K. Goh, K.C. Tan, An investigation on noisy environments in evolutionary

multiobjective optimization, IEEE Trans. Evolut. Comput. 11 (3) (2007) 354–381.
[78] C.K. Goh, K.C. Tan, Noise handling in evolutionary multi-objective optimization, Proc. IEEE Congr. Evolut. Comput. (2006) 1354–1361.
[79] C.K. Goh, K.C. Tan, Evolutionary multi-objective optimization in uncertain environments: issues and Algorithms, Stud. Comput. Intell. 186 (2009).
[80] C.K. Goh, K.C. Tan, C.Y. Cheong, Y.S. Ong, An investigation on noise-induced features in robust evolutionary multi-objective optimization, Expert Syst. Appl. 37 (8) (2010) 5960–5980.
[81] D. E.Goldberg, R. E.Smith, Nonstationary function optimization using genetic algorithms with dominance and diploidy, Genetic Algorithms And Their Application: in: Proceedings of the Second International Conference On Genetic Algorithms, 1987, pp. 59–68
[82] D.E. Goldberg, K. Deb, J.H. Clark, Genetic algorithms, noise, and the sizing of populations, Complex Syst. 6 (1992) 333–362.
[83] G. Gopalakrishnan, B. Minsker, D. Goldberg, Optimal sampling in a noisy genetic algorithm for risk-based remediation design, J. Hydroinformatics 5 (1) (2003) 11–25.
[84] J.J. Grefenstette, Genetic algorithms for changing environments, Proceedings of Parallel Problem Solving from Nature 2, Springer Berlin Heidelberg, 1992, pp. 137–144 (1992).
[85] U. Hammel, T. Bäck, Evolution strategies on noisy functions how to improve convergence properties, Proc. Parallel Probl. Solving Nat., Springe. Berl. Heidelb. (1994) 159–168.
[86] N. Hansen, A.S.P. Niederberger, L. Guzzella, P. Koumoutsakos, A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion, IEEE Trans. Evolut. Comput. 13 (1) (2009) 180–197.
[87] R. Hooke, T.A. Jeeves, Direct search solution of numerical and statistical problems, J. ACM 8 (2) (1961) 212–229.
[88] X. Hu, R.C. Eberhart, Adaptive particle swarm optimization: detection and response to dynamic systems, Proc. IEEE Congr. Evolut. Comput. (2002) 1666–1670.
[89] E.J. Hughes, Constraint handling with uncertain and noisy multi-objective evolution, Proc. IEEE Congr. Evolut. Comput. 2 (2001) 963–970.
[90] E.J. Hughes, Evolutionary algorithm with a novel insertion operator for optimising noisy functions, Proc. IEEE Congr. Evolut. Comput. 1 (2000) 790–797.
[91] E.J. Hughes, Evolutionary multi-objective ranking with uncertainty and noise, in: Proceedings of Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, 2001, pp. 329–343.
[92] G. Iacca, F. Neri, E. Mininno, Noise analysis compact differential evolution, Int. J. Syst. Sci. 43 (7) (2012) 1248–1267.
[93] H. Jang, R. Choe, K.R. Ryu, Deriving a robust policy for container stacking using a noise-tolerant genetic algorithm, Proc. ACM Res. Appl. Comput. Symp. (2012) 31–36.
[94] S. Janson, M. Middendorf, A hierarchical particle swarm optimizer for noisy and dynamic environments, Genet. Program. Evol. Mach. 7 (4) (2006) 329–354.
[95] M. Jebalia, A. Auger, On the convergence of the (1+1)-ES in noisy spherical environments, Evolution Artificielle, Oct 2007, Tours, France. 2007
[96] Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation, Soft Comput. 9 (1) (2005) 3–12.
[97] Y. Jin, B. Sendhoff, Fitness approximation in evolutionary computation-A survey, Proc. Genet. Evolut. Comput. Conf. (2002) 1105–1112.
[98] Y. Jin, J. Branke, Evolutionary optimization in uncertain environments-a survey, IEEE Trans. Evolut. Comput. 9 (3) (2005) 303–317.
[99] Y. Jin, M. Olhofer, B. Sendhoff, A framework for evolutionary optimization with approximate fitness functions, IEEE Trans. Evolut. Comput. 6 (5) (2002) 481–494.
[100] Y. Jin, M. Olhofer, B. Sendhoff, On evolutionary optimization with approximate fitness functions, Proc. Genet. Evolut. Comput. (2000) 786–793.
[101] W.D. Kelton, A.M. Law, Simulation modeling and analysisBoston, McGraw Hill, 2000.
[102] J. Kennedy, Bare bones particle swarms, Proc. Swarm Intell. Symp. (2003) 80–87.
[103] J. Kiefer, Sequential minimax search for a maximum, Proc. Am. Math. Soc. 4 (3) (1953) 502–506.
[104] H. Kita, Y. Sano, Genetic algorithms for optimization of uncertain functions and their applications, Proc. SICE Annu. Conf. 3 (2003) 2744–2749.
[105] J. Knowles, D. Corne, A. Reynolds, Noisy multiobjective optimization on a budget of 250 evaluations, Proc. Evolut. Multi-Criterion Optim., Springe. Berl. Heidelb. (2009) 36–50.
[106] D.E. Knuth, Semin. Algorithms, Art. Comput. Program. 2 (1981).
[107] T. Krink, B. Filipič, G.B.Fogel, Noisy optimization problems-a particular challenge for differential evolution? in: Proceedings of IEEE Congress on Evolutionary Computation, vol. 1, 2004, pp. 332–339. IEEE
[108] J.W. Kruisselbrink, E. Reehuis, A. Deutz, T. Bäck, M. Emmerich, Using the uncertainty handling CMA-ES for finding robust optimaProceedings of Genetic and evolutionary computation, ACM, 2011, pp. 877–884.
[109] K.H. Lee, G.J. Park, Robust optimization considering tolerances of design variables, Comput. Struct. 79 (1) (2001) 77–86.
[110] L.H. Lee, E.P. Chew, A simulation study on sampling and selecting under fixed computing budget, in: Proceedings of the IEEE Winter Simulation Conference, vol. 1, 2003, pp. 535–542.
[111] P. Legendre, D. Borcard, Statistical comparison of univariate tests of homogeneity of variances, Journal of Statistical Computation and Simulation. Département de sciences biologiques, Université de Montréal, vol. 514, 2000.
[112] B. Levitan, S. Kauffman, Adaptive walks with noisy fitness measurements, Mol. Divers. 1 (1) (1995) 53–68.
[113] X. Li, Niching without niching parameters: particle swarm optimization using a ring topology, IEEE Trans. Evolut. Comput. 14 (1) (2010) 150–169.
[114] K.H. Liang, X. Yao, C. Newton, Combining landscape approximation and local search in global optimization, Proc. IEEE Congr. Evolut. Comput. 2 (1999) 1520.
[115] P. Limbourg, D.E.S. Aponte, An optimization algorithm for imprecise multi-objective problem functions, Proc. IEEE Congr. Evolut. Comput. 1 (2005) 459–466.
[116] B. Liu, X. Zhang, H. Ma, Hybrid differential evolution for noisy optimization, Proc. IEEE Congr. Evolut. Comput. (2008) 587–592.
[117] C. Liu, Chun-an, New dynamic constrained optimization PSO algorithm, Proc. Nat. Comput. 7 (2008) 650–653.
[118] J. Liu, D.L.S. Pierre, O. Teytaud, A mathematically derived number of resamplings for noisy optimization, in: Proceedings of the Companion Publication of Annual Conference on Genetic and Evolutionary Computation, ACM, 2014, pp. 61–62.
[119] J. Liu, M. Fairbank, D.P. Liébana, S.M. Lucas, (*arXiv preprint arXiv*)Optim. resampling noisy OneMax Probl., 06641 (1607) 2016.
[120] J. Lozano, Towards a new evolutionary computation: advances on estimation of distribution algorithms 192, Springer Science & Business Media, 2006.
[121] H. Ma, M. Fei, D. Simon, Z. Chen, Biogeography-based optimization in noisy environments, Trans. Inst. Meas. Control 37 (2) (2015) 190–204.
[122] E.D. Mario, I. Navarro, A. Martinoli, Analysis of fitness noise in particle swarm optimization: from robotic learning to benchmark functions, Proc. IEEE Congr. Evolut. Comput. (2014) 2785–2792.
[123] S. Markon, D.V. Arnold, T. Back, T. Beielstein, H.G. Beyer, Thresholding-a selection operator for noisy ES, Proc. IEEE Congr. Evolut. Comput. 1 (2001) 465–472.
[124] E. Mendel, R.A. Krohling, M. Campos, Swarm algorithms with chaotic jumps applied to noisy optimization problems, Inf. Sci. 181 (20) (2011) 4494–4514.
[125] J.M. Mendel, R.I. John, F. Liu, Interval type-2 fuzzy logic systems made simple, IEEE Trans. Fuzzy Syst. 14 (6) (2006) 808–821.
[126] M.H.S. Mendes, G.L. Soares, J. Coulomb, J.A. Vasconcelos, A surrogate genetic programming based model to facilitate robust multi-objective optimization: a case study in Magnetostatics, IEEE Trans. Magn. 49 (5) (2013) 2065–2068.
[127] J. J.Merelo, F. Liberatore, A.F. Ares, R. Garcia, Z. Chelly, C. Cotta, N. Rico, A.M. Mora, P.G.Sanchez, There is noisy lunch: A study of noise in evolutionary optimization problems, in: Proceedings of International Joint Conference on Computational Intelligence,2015, pp. 261–268.
[128] J.J. Merelo, Z. Chelly, A. Mora, A.F. Ares, A.I.E. Alcázar, C. Cotta, P.D.L. Cuevas, N. Rico, A statistical approach to dealing with noisy fitness in evolutionary algorithms, *Proc. Comput. Intell.*, Springe. Int. Publ. (2016) 79–95.
[129] Z. Michalewicz, Genetic algorithms+data structures=evolution programs, Springer Science & Business Media, 2013.
[130] B.L. Miller, D.E. Goldberg, Genetic algorithms, selection schemes, and the varying effects of noise, Evolut. Comput. 4 (2) (1996) 113–131.
[131] B.L. Miller, Noise, sampling, and efficient genetic algorithms, Ph. D. Thesis, Dep. Comput. Sci., Univ. Ill. Urbana-Champaign, TR 97001 (1997).
[132] E. Mininno, F. Neri, A memetic differential evolution approach in noisy optimization, Memetic Comput. 2 (2) (2010) 111–135.
[133] S.A. Morales, M.L. Cauwet, J. Liu, O. Teytaud, Simple and cumulative regret for continuous noisy optimization, Theor. Comput. Sci. 617 (2016) 12–27.
[134] S. A. Morales, M. L. Cauwet, O. Teytaud, Evolution strategies with additive noise: A convergence rate lower bound, in: Proceedings of ACM Conference on F*oundations of Genetic Algorithms XIII*, ACM, 2015, pp. 76–84
[135] S.A. Morales, M.L. Cauwet, O. Teytaud, Analysis of different types of regret in continuous noisy optimization, in: Proceedings of Genetic and Evolutionary Computation Conference, ACMpp. 205–212, 2016.
[136] R.W. Morrison, Designing evolutionary algorithms for dynamic environments, Springer Science & Business Media, 2013.
[137] R. Narducci, B. Grossman, M. Valorani, A. Dadone, R.T. Haftka, Optimization methods for non-smooth or noisy objective functions in fluid design problems, AIAA Pap. (1648) 1995.
[138] K.S. Narendra, M.L.A.A. Thathachar, Learning automata-a survey, IEEE Trans. Syst., Man Cybern. 4 (1974) 323–334.
[139] V. Nissen, J. Propach, On the robustness of population-based versus point-based optimization in the presence of noise, IEEE Trans. Evolut. Comput. 2 (3) (1998) 107–119.
[140] V. Nissen, J. Propach, Optimization with noisy function evaluationsProceedings of Parallel Problem Solving from Nature, Springer Berlin Heidelberg, 1998, pp. 159–168.
[141] Y.S. Ong, P.B. Nair, K. Lum, Max-min surrogate-assisted evolutionary algorithm for robust design, IEEE Trans. Evolut. Comput. 10 (4) (2006) 392–404.
[142] I. Paenke, J. Branke, Y. Jin, Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation, IEEE Trans. Evolut. Comput. 10 (4) (2006) 405–420.
[143] L. Painton, Laura, U. Diwekar, Stochastic annealing for synthesis under uncertainty, Eur. J. Oper. Res. 83 (3) (1995) 489–502.
[144] K. Pal, C. Saha, S. Das, C.A. Coello Coello, Dynamic constrained optimization with offspring repair based gravitational search algorithm, Proc. IEEE Congr. Evolut. Comput. (2013) 2414–2421.
[145] H. Pan, L. Wang, B. Liu, Particle swarm optimization for function optimization in noisy environment, Appl. Math. Comput. 181 (2) (2006) 908–919.
[146] T. Park, K. R. Ryu, Accumulative sampling for noisy evolutionary multi-objective optimization, in: Proceedings of the ACM 13th annual conference on Genetic and evolutionary computation, 2011, pp. 793–800
[147] M. Pelikan, K. Sastry, E.C. Paz (Eds.), Scalable optimization via probabilistic modeling: From algorithms to applications, vol. 33, 2007, Springer.
[148] D.H. Phan, J. Suzuki, A non-parametric statistical dominance operator for noisy

multiobjective optimizationSimulated Evolution and Learning, Springer Berlin Heidelberg, 2012, pp. 42–51.
[149] A.D. Pietro, L. While, L. Barone, Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions, Proc. IEEE Congr. Evolut. Comput. 2 (2004) 1254–1261.
[150] A.D. Pietro, Optimising evolutionary strategies for problems with varying noise strength, Ph. D. Thesis, University of Western Australia, 2007.
[151] K. Price, R.M. Storn, J.A. Lampinen, Differential evolution: a practical approach to global optimization, Springer Science & Business Media, 2006.
[152] S. Rahnamayan, H.R. Tizhoosh, M. Salama, Opposition-based differential evolution for optimization of noisy problems, Proc. IEEE Congr. Evolut. Comput. (2006) 1865–1872.
[153] P. Rakshit, A. Konar, P. Bhowmik, I. Goswami, S. Das, L.C. Jain, A.K. Nagar, Realization of an adaptive memetic algorithm using differential evolution and Q-learning: a case study in multirobot path planning, IEEE Trans. Syst., Man, Cybern.: Syst. 43 (4) (2013) 814–831.
[154] P. Rakshit, A. Konar, S. Das, L.C. Jain, A.K. Nagar, Uncertainty management in differential evolution induced multiobjective optimization in presence of measurement noise, IEEE Trans. Syst., Man, Cybern.: Syst. 44 (7) (2014) 922–937.
[155] P. Rakshit, A. Konar, Differential evolution for noisy multiobjective optimization, Artif. Intell. 227 (2015) 165–189.
[156] P. Rakshit, A. Konar, Extending multi-objective differential evolution for optimization in presence of noise, Inf. Sci. 305 (2015) 56–76.
[157] P. Rakshit, A. Konar, Non-dominated Sorting Bee Colony optimization in the presence of noise, Soft Comput. 20 (3) (2016) 1139–1159.
[158] P. Rakshit, A. Konar, Atulya K. Nagar, Artificial bee colony induced multi-objective optimization in presence of noise, Proc. IEEE Congr. Evolut. Comput. (2014) 3176–3183.
[159] P. Rakshit, A. Konar, Atulya K. Nagar, Type-2 fuzzy induced non-dominated sorting bee colony for noisy optimization, Proc. IEEE Congr. Evolut. Comput. (2015) 3176–3183.
[160] S. Rana, L.D. Whitley, R. Cogswell, Searching in the presence of noise, Proc. Parallel Probl. Solving Nat., Springe. Berl. Heidelb. (1996) 198–207.
[161] M. Rattray, J. Shapiro, Noisy fitness evaluation in genetic algorithms and the dynamics of learning, Proc. Found. Genet. Algorithms (1997) 117–139.
[162] T. Ray, Constrained robust optimal design using a multiobjective evolutionary algorithm, Proc. IEEE Congr. Evolut. Comput. 1 (2002) 419–424.
[163] I. Rechenberg, Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, frommann-holzbog, Stuttgart, 1973.
[164] C. W. Reynolds, Evolution of corridor following behavior in a noisy world, From Animals to Animats 3, in: Proceedings of the Third International Conference on Simulation of Adaptive Behavior, 1994, pp. 402–410
[165] G. Rudolph, A partial order approach to noisy fitness functions, Proc. IEEE Congr. Evolut. Comput. 1 (2001) 318–325.
[166] S. Russell, P. Norvig, Artificial Intelligence: a modern approach, Egnlewood Cliffs 25, Prentice-Hall, 1995.
[167] Y. Sano, H. Kita, Optimization of noisy fitness functions by means of genetic algorithms using history of search with test of estimation, Proc. IEEE Congr. Evolut. Comput. 1 (2002) 360–365.
[168] Y. Sano, H. Kita, Optimization of noisy fitness functions by means of genetic algorithms using history of search, Proc. Parallel Probl. Solving Nat., Springe. Berl. Heidelb. (2000) 571–580.
[169] Y. Sano, H. Kita, I. Kamihira, M. Yamaguchi, Online optimization of an engine controller by means of a genetic algorithm using history of search, in: Proceedings of the 26th Annual Conference of IEEE Industrial Electronics Society, vol. 4pp. 2929–2934, 2000.
[170] D. Sheskin, Handbook of parametric and nonparametric statistical procedures, CRC Press, 2003.
[171] L. Shi, K. Rasheed, ASAGA: an adaptive surrogate-assisted genetic algorithmProceedings of Genetic and Evolutionary Computation, ACM, 2008, pp. 1049–1056.
[172] V.A. Shim, K.C. Tan, J.Y. Chia, A. Al Mamun, Multi-objective optimization with estimation of distribution algorithm in a noisy environment, Evolut. Comput. 21 (1) (2013) 149–177.
[173] F. Siegmund, Sequential sampling in noisy multi-objective evolutionary optimization, Master's Thesis, University of Skövde, School of Humanities and Informatics, 2009.
[174] F. Siegmund, A.H.C. Ng, K. Deb, A comparative study of dynamic resampling strategies for guided evolutionary multi-objective optimization, Proc. IEEE Congr. Evolut. Comput. (2013) 1826–1835.
[175] F. Siegmund, A.H.C. Ng, K. Deb, Hybrid dynamic resampling for guided evolutionary multi-objective optimizationProceedings of Evolutionary Multi-Criterion Optimization, Springer International Publishing, 2015, pp. 366–380.
[176] F. Siegmund, A.H.C. Ng, K. Deb, Hybrid dynamic resampling algorithms for evolutionary multi-objective optimization of invariant-noise problemsProceedings of European Conference on the Applications of Evolutionary Computation, Springer International Publishing, 2016, pp. 311–326.
[177] F. Siegmund, A.H.C. Ng, K. Deb, Dynamic resampling for preference-based evolutionary multi-objective optimization of stochastic systems, in: Proceedings of International Conference on Multiple Criteria Decision Making, Hamburg, Germany, 2015, COIN Report Number 2015020, pp. 1–24
[178] F. Siegmund, A.H.C. Ng, K. Deb, Standard error dynamic resampling for preference-based evolutionary multi-objective optimization, submitted to Computational Optimization and Innovation Laboratory, 2016, COIN Report Number 2015021, pp. 1–13.
[179] F. Siegmund, A.H.C.Ng, K.Deb, A ranking and selection strategy for preference-based evolutionary multi-objective optimization of variable-noise problems,submitted toComputational Optimization and Innovation Laboratory, 2016, COIN Report Number COIN Report Number002, 2016, pp. 1–13.
[180] D. Simon, Biogeography-based optimization, IEEE Trans. Evolut. Comput. 12 (6) (2008) 702–713.
[181] D. Simon, M. Ergezer, D. Du, R. Rarick, Markov models for biogeography-based optimization, IEEE Trans. Syst., Man, Cybern., Part B: Cybern. 41 (1) (2011) 299–306.
[182] A. Singh, Uncertainty based multi-objective optimization of groundwater remediation design, Master's Thesis, University of Illinois at Urbana-Champaign, 2003.
[183] L. Siwik, S. Natanek, Elitist evolutionary multi-agent system in solving noisy multi-objective optimization problems, Proc. IEEE Congr. Evolut. Comput. (2008) 3319–3326.
[184] G.L. Soares, F.G. Guimarães, C.A. Maia, J.A. Vasconcelos, L. Jaulin, Interval robust multi-objective evolutionary algorithm, in: Proceedings of IEEE Congress on Evolutionary Computation, 2009, pp. 1637–1643.
[185] G.L. Soares, R.L.S. Adriano, C.A. Maia, L. Jaulin, J.A. Vasconcelos, Robust multi-objective TEAM 22 problem: a case study of uncertainties in design optimization, IEEE Trans. Magn. 45 (3) (2009) 1028–1031.
[186] P. Stagge, Averaging efficiently in the presence of noiseProceedings of Parallel Problem Solving from Nature, Springer Berlin Heidelberg, 1998, pp. 188–197.
[187] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.
[188] P.D. Stroud, Kalman-extended genetic algorithm for search in nonstationary environments with noisy fitness evaluations, IEEE Trans. Evolut. Comput. 5 (1) (2001) 66–77.
[189] A. Syberfeldt, A. Ng, R.I. John, P. Moore, Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling, Eur. J. Oper. Res. 204 (3) (2010) 533–544.
[190] K.C. Tan, T.H. Lee, E.F. Khor, Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization, IEEE Trans. Evolut. Comput. 5 (6) (2001) 565–588.
[191] J. Teich, Pareto-front exploration with uncertain objectives, in: Proceedings of Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, 2001, pp. 314–328.
[192] O. Teytaud, A. Auger, On the adaptation of noise level for stochastic optimization,, Proc. IEEE Congr. Evolut. Comput. (2007) 3027–3034.
[193] H. Trautmann, J. Mehnen, B. Naujoks, Pareto-dominance in noisy environments, Proc. IEEE Congr. Evolut. Comput. (2009) 3119–3126.
[194] A. Tsoularis, Analysis of logistic growth models, Math. Biosci. 179 (1) (2002) 21–55.
[195] S. Tsutsui, A. Ghosh, Y. Fujimoto, A robust solution searching scheme in genetic search,, *Proc. Parallel Probl. Solving Nat.*, Springe. Berl. Heidelb. (1996) 543–552.
[196] J.R. Vilela, Population statistics for particle swarm optimization on problems subject to noise, Ph. D. Thesis, Victoria University of Wellington, 2014.
[197] C.Villa, E.Lozinguez, R.Labayrade, Multi-objective optimization under uncertain objectives: application to engineering design problem, in: Proceedings of Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, 2013, pp. 796–810.
[198] D. Wiesmann, U. Hammel, T. Bäck, Robust design of multilayer optical coatings by means of evolutionary algorithms, IEEE Trans. Evolut. Comput. 2 (4) (1998) 162–167.
[199] S. Yang, Associative memory scheme for genetic algorithms in dynamic environments, in: Proceedings of Applications of Evolutionary Computing, Springer Berlin Heidelberg, 2006, pp. 788–799.
[200] J.Q. Zhang, L.W. Xu, J. Ma, M.C. Zhou, A learning automata-based particle swarm optimization algorithm for noisy environment, IEEE Congr. Evolut. Comput. (2015) 141–147.
[201] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multi-objective Optimization Test Instances for the CEC 2009 Special Session and Competition, Working Report, CES-887, School of Computer Science and Electrical Engineering, University of Essex, 2008.
[202] Z. Zhang, T. Xin, Immune algorithm with adaptive sampling in noisy environments and its application to stochastic optimization problems, IEEE Comput. Intell. Mag. 2 (4) (2007) 29–40.
[203] E. Zitzler, K. Deb, L. Thiele, Comparison of multi-objective evolutionary algorithms: empirical results, J. Evolut. Comput. 8 (2000) 173–195.